



***Facultad
de
Ciencias***

**Búsqueda de una sobredensidad de
galaxias a alto corrimiento hacia el rojo,
usando imágenes tomadas con
filtros sintonizables.
(Search for an overdensity of galaxies at high
redshift, using images taken with tunable
filters.)**

**Trabajo de Fin de Grado
para acceder al**

GRADO EN FÍSICA

Autor: María Ruiz Pérez

Director: Dr. Francisco Jesús Carrera Troyano

Julio - 2019

Índice general

1. Introducción	6
1.1. Objetivos del trabajo.	6
1.2. Marco teórico	6
1.3. Espectro de galaxias observadas a alto <i>redshift</i> .	8
1.4. ¿Cómo detectar una sobredensidad de galaxias?	
La técnica seguida.	10
2. Datos.	11
2.1. Cámaras CCD.	11
2.2. Filtros sintonizables.	11
2.3. Origen de los datos: El GTC	12
2.4. OSIRIS	12
2.5. Base de datos: SDSS Sloan Digital Sky Survey	13
2.5.1. El cuásar.	14
2.5.2. La estrella de calibración.	15
3. Metodología.	17
3.1. Concepto de flujo astronómico.	17
3.2. Fase 1: Detección de fuentes y extracción de datos de las imágenes astronómicas.	18
3.3. Fase 2: Calibración fotométrica y análisis de la información.	19
3.4. Descentrado de los filtros.	21
3.4.1. Midiendo el <i>redshift</i> .	22
3.5. Forma de obtener resultados con los datos disponibles.	23
3.6. Señal/ruido	24
4. Resultados	25
4.1. Fase 3: Método de selección y su descripción física.	25
4.2. Selección de posibles candidatas.	30
5. Discusión	31
5.1. Limitaciones iniciales.	31
5.2. Fuentes seleccionadas.	31
5.3. Caso I: La mejor selección.	32
5.3.1. La mejor candidata: 303.	32
5.3.2. Candidata aceptable: 401.	34

5.3.3. Cumplimiento del método de selección pero no de los criterios físicos: 58 y 623.	36
5.4. Limitaciones encontradas tras el análisis.	39
5.5. Tamaño físico del área observada frente al de un cúmulo.	40
5.6. Comparación con los resultados en submilimétricas.	40
6. Conclusiones y trabajo futuro.	42
Bibliografía	42
Apéndice A: Valores observados de los parámetros de selección.	46
Apéndice B: Descripción breve de programas y scripts. Y códigos creados.	49

Agradecimientos

Con la perspectiva que te da la calma tras la tempestad, tengo claro pese a mis momentos pasados de dudas, que el haber estudiado Física me ha merecido la pena.

Va a ser y es inevitable para mi el recordar mis años como estudiante de Física, como una etapa en la que la enfermedad y los problemas adversos me han puesto al límite, convirtiendo la preparación de mis exámenes como el menor de mis problemas. Es por ello que mi TFG lo he querido afrontar como un regalo. Pues para mi este TFG me ha permitido experimentar en ser durante unas cuantas semanas, aquello a lo que de pequeña jugué durante años ser: astrónoma.

Es en este punto donde me acuerdo del Dr. Francisco J. Carrera Troyano, que aunque él a mi no me conocía, yo a él si. Y le doy las gracias por permitirme inmiscuirme en una investigación real y por las charlas sobre física que he tenido con él. Espero haber estado a la altura como aprendiz.

También he de agradecer la labor que mi doctora Rafaela Mardarás ha realizado durante gran parte de mi vida por mí. Y la agradeceré el enfadarse conmigo ante los escasos aunque intensos, momentos de querer abandonar la carrera, que ahora termino con este trabajo.

A mis amigos, los de siempre y a los que gracias a estudiar Física he conseguido.

Y cómo no, mi agradecimiento a mi madre no solo por servirme de atril para mis apuntes en las salas de espera del hospital, sino por acarrear mis miedos dejando a un lado los suyos.

Gracias.

Resumen

En este trabajo se persigue investigar sobre un exceso de galaxias, una sobredensidad, en torno a un cuásar a alto *redshift*, $z = 5.04$. La detección de estas galaxias es de especial relevancia puesto que se tratan de galaxias muy lejanas, $z > 5$, y por tanto, proporcionan información sobre la formación de las estructuras que vemos hoy día en el Universo. Por esto, son extremadamente difíciles de detectar y conseguirlo supone todo un reto. Además se presenta un método para alcanzar tal desafío. Los objetivos de este trabajo son, por tanto, el método desarrollado y la detección de alguna de estas galaxias.

Para llevar a cabo el estudio, se ha trabajado con imágenes astronómicas tomadas en el GTC por medio de filtros sintonizables, siendo el uso de estos filtros algo novedoso en este tipo de trabajo.

El método de selección desarrollado, se basa en la presencia de la absorción y emisión de líneas procedentes de átomos de hidrógeno en los espectros de las galaxias a alto z . Haciendo uso de este método se ha pasado de una muestra de 986 fuentes detectadas a una muestra de 40, de las cuales a su vez se han encontrado dos posibles buenas candidatas a estar relacionadas con el cuásar.

Finalmente se comparan las galaxias candidatas encontradas en este trabajo, con las que fueron encontradas en longitudes de onda submilimétricas (21) en torno al mismo cuásar.

El hecho de que el número de galaxias encontradas sea bajo, en base a la comparación realizada, no permite confirmar la existencia de la sobredensidad. Esto se piensa que puede deberse, además a porque el método sea en primera aproximación, a los defectos de la configuración observacional y a un cociente señal/ruido de las observaciones bajo.

Palabras clave: Universo, cuásar, galaxias, sobredensidad, filtros sintonizables.

Abstract

In this work we seek to look for excess galaxies, an overdensity, around a quasar at high redshift, $z = 5.04$. The detection of these galaxies is of special relevance since they are very distant galaxies, $z > 5$, and therefore , provide information on the formation of the structures that we see today in the Universe.

Because of this, they are extremely difficult to detect and to achieve this is quite a challenge. In addition, a method to achieve such a challenge is presented. The objectives of this work, therefore are, the developed method and the detection of some of these galaxies.

To carry out the study, we have worked with astronomical images taken in the GTC by means of tunable filters, being the use of these filters something new in this type of work.

The method of selection developed, is based on the presence of the absorption and emission by hydrogen atoms in the spectra of the galaxies at high z . Using this method, a sample of 986 detected sources reduced into a sample of 63, of which in turn two possible good candidates have been found to be related to quasar.

Finally, the candidate galaxies found in this work are compared with those found in sub-millimeter wavelengths (17) around the same quasar.

The fact that the number of galaxies found is low, based on the comparison made, does not confirm the existence of overdensity. This is thought to be due, in addition to the method being a first approximation, to the defects of the observational configuration and to a signal/noise ratio of the low observations.

Keywords: Universe, quasar, galaxies, overdensity, tuneable filters.

Capítulo 1

Introducción

En esta sección se establece el marco de trabajo en el que se desarrolla el estudio. Además se introducen las bases teóricas sobre las que versa el resto del trabajo. Y se hace un resumen de los elementos principales para la adquisición de las imágenes astronómicas que en él se estudian.

1.1. Objetivos del trabajo.

La motivación de este trabajo radica en aumentar la comprensión de la formación de las grandes estructuras que pueblan el Universo que vemos hoy en día. Para ello hay que estudiar ambientes a un alto desplazamiento al rojo o *redshift*, en inglés, y denotado con z , (en lo que sigue se utilizará *redshift* por brevedad). Este trabajo se centra en particular en el entorno de un cuásar a $z=5.06$, en el cual se ha detectado una sobredensidad de galaxias en longitudes de onda submilimétricas.

El objetivo principal de este trabajo es investigar esta sobredensidad realizando la búsqueda de galaxias a alto *redshift* mediante una técnica basada en la emisión y absorción de Lyman alpha. Y consiste en la detección y tratamiento de fuentes encontradas asegurándose de que no hay ninguna detección espuria (son aquellas detecciones a las que no hay un objeto físico asociado (13)).

Un segundo objetivo de este trabajo es desarrollar un método para la selección de las galaxias que se buscan de entre todas las fuentes detectadas. Y consiste en el diseño del método de selección y la aplicación del mismo para conseguir el objetivo principal. Posteriormente, se hace una selección de las fuentes encontradas para indagar sobre la existencia, o no, de una sobredensidad en torno al cuásar.

1.2. Marco teórico

Aunque el universo es homogéneo e isótropo a gran escala, esto es en escalas de centenares de megaparsecs (20), en escalas del orden de megaparsecs ya no lo es, Fig. (1.1), y las galaxias se agrupan por efectos gravitatorios en cúmulos de galaxias (12), en cuya formación intervienen también una serie de procesos físicos aún hoy por entender. Estos cúmulos que vemos hoy en día en el Universo se forman a partir de sobredensidades de materia hacia donde gravitan las galaxias, y aún se están formando. Encontrar estas sobredensidades a alto *redshift* tiene especial relevancia porque son los progenitores, de los cúmulos ricos que vemos en la actualidad (6), (11).

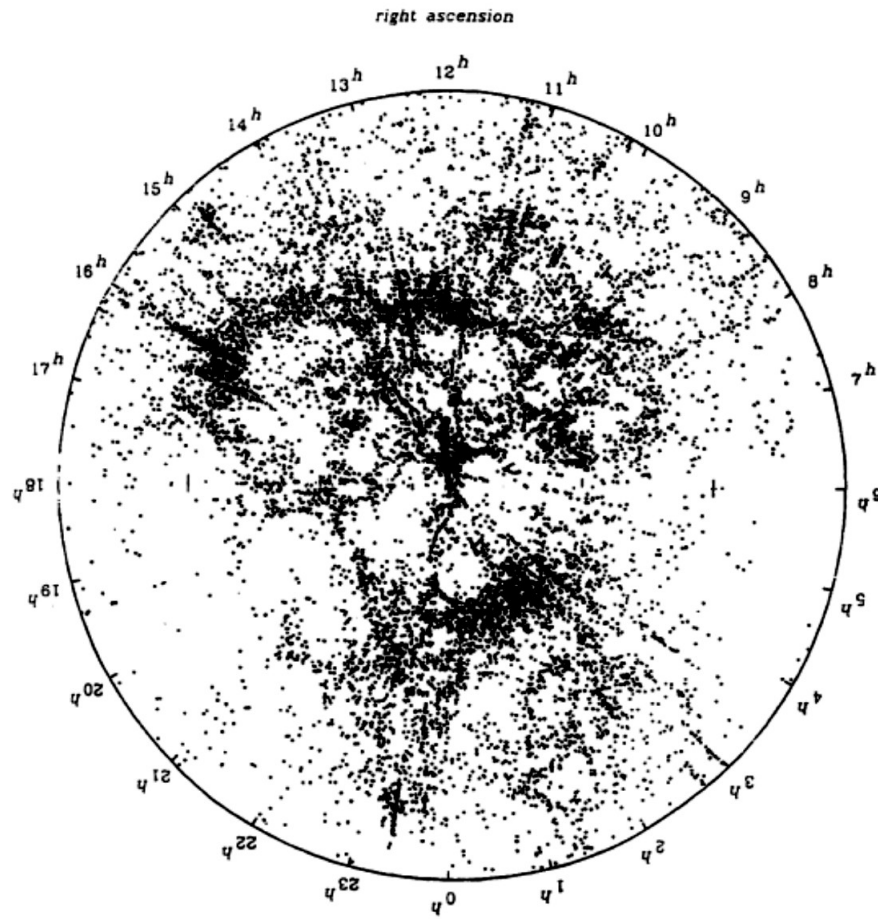


Figura 1.1: *Distribución de galaxias en el Universo cercano, obtenida en el estudio realizado por el centro de Astrofísica de Harvard-Smithsonian. El mapa contiene más de 14000 galaxias que forman una muestra estadística completa alrededor del cielo entre declinaciones de 8.5° y 44.5° . Todas las galaxias tienen velocidades de recesión inferiores a 15000 km s^{-1} . Nuestra galaxia se encuentra en el centro del mapa y el radio del círculo delimitador es de $150 h^{-1} \text{ Mpc}$. Las galaxias dentro de esta porción se han proyectado en un plano para mostrar las características a gran escala en la distribución de galaxias. La distribución de galaxias es altamente irregular con grandes agujeros, filamentos y cúmulos de galaxias en todo el universo local.*(14)

Los cuásares son núcleos de galaxias activas y son los objetos persistentes más brillantes en el Universo. Y, por tanto, pueden ser detectados a grandes distancias (8) lo que implica a tiempos muy tempranos.

En torno a algunos de estos cuásares se han encontrado sobredensidades, en concreto de galaxias en ondas submilimétricas (21). Esto es un indicador muy bueno ya que las galaxias que están formando estrellas, calientan el polvo en torno a las estrellas en formación, el cual emite térmicamente en el IR lejano. Y cuando estas fuentes están a alto z esta emisión se observa en ondas submilimétricas. Luego, si se detecta una sobredensidad de galaxias en submilimétricas en torno a un cuásar, es un buen indicador de formación estelar y por tanto de la existencia de galaxias que están creciendo. Por contra, a partir de la emisión en submilimétricas no se puede saber el *redshift*.

Los cuásares son por tanto buenos indicadores de los cúmulos de galaxias en el Universo (12). Entonces, la búsqueda de cuásares es un método bueno en principio para localizar sitios con grandes densidades de galaxias a alto *redshift*.

En este trabajo se quiere comprobar, mediante una técnica basada en la emisión y absorción en Lyman- α , si además de en ondas submilimétricas, también podemos detectar una sobredensidad de galaxias en torno a un cuásar en el óptico y al mismo z que dicho cuásar. Para ello se ha observado la zona en torno al cuásar SDSS J075618.13+410408.6 a $z=5.06$.

Lo diferente en este trabajo frente a otros trabajos de búsqueda de fuentes a alto *redshift* es la técnica. Porque aunque este trabajo se basa en el mismo principio físico que otros, las imágenes astronómicas con las que se trabaja durante este trabajo han sido realizadas mediante el uso de filtros sintonizables cuyo uso para este tipo de investigaciones no está actualmente extendido. Esto dota al trabajo de un carácter novedoso.

Además, se quiere hacer notar al lector de el marco de trabajo de este trabajo, $z > 5$. Este *redshift*, tan elevado, supone remontarse a una época en la que el Universo tenía el 10 % (29) de la edad actual. Esto significa que una estrella como el Sol (que estimadamente va a vivir 10000 millones de años) no le hubiera dado tiempo a evolucionar desde el Big Bang, sería prácticamente recién nacida. Así pues, durante este trabajo se ha explorado una época del Universo muy interesante pero muy compleja.

1.3. Espectro de galaxias observadas a alto *redshift*.

La radiación que viene de las galaxias a alto *redshift* tiene que atravesar mucha cantidad de material, del medio intergaláctico, entre las galaxias y nosotros. El elemento más abundante en el Universo es el hidrógeno (24).

Los átomos de hidrógeno del medio intergaláctico absorben la radiación de estos objetos a alto *redshift*. Como consecuencia se encuentran características espectrales debidas al hidrógeno, en el espectro de esas galaxias. Estas características pueden observarse también en el espectro del cuásar (3) que se está tratando en el presente trabajo, ver Fig. (1.2) y Fig. (2.3).

El efecto más llamativo, es la absorción prácticamente total del espectro hacia el azul de $91.2(1+z)$ nm, donde 91.2 nm es el denominado *Límite de Lyman*. Y lo que se observa en el espectro es una caída brusca a partir de este límite. La causa es que todos los átomos de hidrógeno cuando absorben radiación a una energía igual o superior a la correspondiente a la longitud de onda del *límite de Lyman*, se ionizan completamente. Como consecuencia se produce un continuo, ya que la absorción se produce para todas las longitudes de onda a partir del límite.

Otra característica del espectro de más relevancia para este trabajo como consecuencia de la absorción producida por los átomos de hidrógeno, es la absorción que se da a $121.6(1+z)$ nm, lo que da lugar a una caída hacia el azul, (a la que a lo largo del presente documento se aludirá también como escalón), también importante, solo que en este caso tras la caída el espectro no desaparece completamente como en la anterior. Este efecto es debido a que esos átomos de hidrógeno, absorben la radiación correspondiente a la transición Lyman- α entre $n=1$ y $n=2$ (7), (27).

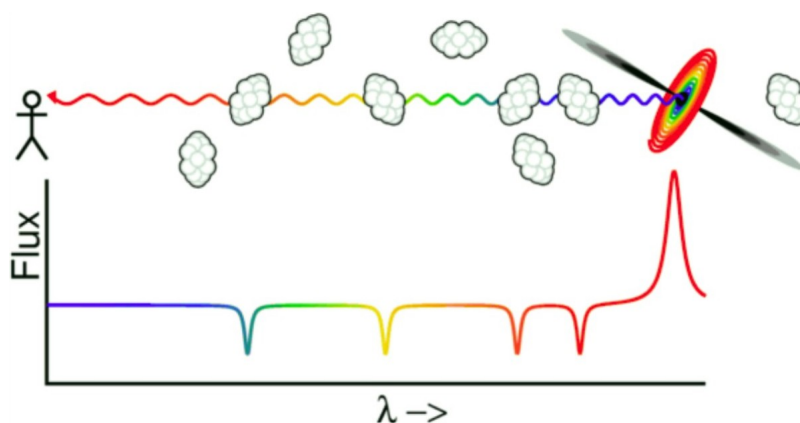


Figura 1.2: Imagen ilustrativa (28) que muestra un cuásar con su línea de emisión Lyman- α desplazada al rojo, y cuatro líneas de absorción de Lyman- α producidas por cuatro de las nubes de hidrógeno, cada una de las cuales se encuentra con la radiación emitida por el cuásar a diferentes distancias y por tanto a distintos corrimientos al rojo.

Hay que decir que cualquier objeto que esté al mismo *redshift* que el del cuásar y en su vecindad, presentará un espectro con la caída descrita. La razón de esto es que el escalón no es debido al objeto en sí, sino al material que la radiación emitida se encuentra por el camino, ver Fig. (1.2).

Por otra parte, los espectros de algunas galaxias presentan líneas de emisión. En concreto la línea Lyman- α se produce cuando el hidrógeno está excitado. Para que una galaxia emita en Lyman- α se necesita mucha cantidad de hidrógeno excitado, lo que requiere que la galaxia tenga una alta formación estelar (o la presencia de un núcleo galáctico activo). Con lo cual, en los espectros de estas galaxias además de la presencia del escalón a $121.6(1+z)$ nm y a 91.2 nm, se observa una línea de emisión a $121.6(1+z)$ nm.

El estudio que compete a este trabajo, se basa en estos fundamentos físicos. Así pues, lo que se busca son por un lado galaxias que presenten una caída brusca de la radiación observada hacia el azul

de la longitud de onda que le corresponda a la línea Lyman- α , que es algo que se busca por ser lo que le ocurre al espectro del cuásar, tal como se observa en la Fig. (2.3). En el presente trabajo estas galaxias serán llamadas *Lyman- α breaks*, *LABs*.

Y por otra parte, en la búsqueda se permite que justo hacia el rojo de donde esté el escalón haya emisión en exceso, ya que se piensa que algunas galaxias en torno al cuásar estén formando estrellas. Estas son entonces las galaxias que además de tener la caída brusca, tengan emisión en Lyman- α , las denominadas *Lyman- α emitters*, *LAEs*.

En definitiva, en este trabajo se buscan galaxias que tengan un escalón en la misma longitud de onda que donde se da para el cuásar. Y esto se va a interpretar como objetos que están a la misma distancia que el cuásar, que se corresponde con la absorción que se observa hacia el azul de Lyman- α . Los filtros sintonizables con los que se realizaron las observaciones de las que se vale este trabajo, están colocados según el diseño inicial, centrados en la emisión de Lyman- α del cuásar.

1.4. ¿Cómo detectar una sobredensidad de galaxias?

La técnica seguida.

La sobredensidad a la que en el presente documento se alude, es un exceso en la densidad de galaxias con respecto al valor promedio global. Las fuentes detectadas son candidatas de las que se toman sus espectros, siendo esto costoso. Por tanto, es necesario hacer una preselección de las galaxias de las que se crea que están al mismo *redshift* que el del cuásar.

La técnica de búsqueda seguida en este trabajo se basa en la absorción y emisión de Lyman- α , sobre imágenes astronómicas realizadas mediante el uso de una serie de filtros sintonizables. Y consiste en buscar galaxias en las que el hidrógeno intergaláctico haya absorbido radiación, desde la línea Lyman- α hacia el azul. Por ello, es por lo que los filtros se pretende que estén centrados en principio, en torno a la línea Lyman- α .

Si se encuentra un exceso de estas galaxias acumuladas en una región del Universo se estará ante una sobredensidad de galaxias. Encontrar una sobredensidad de estas en torno a un cuásar, conlleva sospechar que las galaxias al estar al mismo *redshift* y en la misma región espacial, guarden alguna relación con el cuásar.

Capítulo 2

Datos.

La realización de este trabajo se ha llevado a cabo a partir de ocho imágenes astronómicas proporcionadas a la autora del mismo en formato FITS (en inglés, Flexible Image Transport System (19)) y estando ya reducidas. Tales imágenes fueron tomadas con el Gran Telescopio de Canarias del 27/12/2011 al 18/02/12, en modo servicio, mediante el uso del instrumento OSIRIS. Todas las imágenes tienen en el centro el cuásar SDSS J075618.13+410408.6 .

2.1. Cámaras CCD.

Las cámaras CCD (Charge Coupled Device) son detectores fotoeléctricos usados, en este contexto, para fotometría ¹ astronómica. Esencialmente lo que tiene es una oblea de silicio en la que la recepción de radiación da lugar a que uno o varios electrones pasen de la banda de valencia a la de conducción. Para evitar que estos electrones circulen por la red cristalina, la superficie de la oblea se recubre con unos electrodos en forma de matriz, que generan una serie de pozos de potencial que acumulan estos electrones mientras se recibe la luz (píxeles). Cuando se desea leer la información, se interrumpe la llegada de luz y se mide la carga en cada uno de esos pozos. Esta información se almacena y puede hacerse por filas y columnas, dando así lugar a una matriz donde el valor de cada elemento es proporcional a la cantidad de luz recibida en el pixel correspondiente.

Cuando se coloca una CCD en el foco de un telescopio para grabar una imagen del campo de visión del instrumento, el telescopio actúa como una cámara de alta potencia (16). Sin embargo, los astrónomos a menudo desean realizar mediciones más específicas de la radiación recibida desde el espacio.(16). Por ello a menudo se combina las mediciones fotométricas con filtros.

2.2. Filtros sintonizables.

Se trata de filtros de banda estrecha que permiten escoger el centro y la anchura del rango de longitudes de onda que transmiten. Esta capacidad de elegir de manera tan selectiva la longitud de onda es la característica que los diferencia de los filtros de banda estrecha convencionales. Los filtros sintonizables constan de dos láminas de vidrio extraordinariamente planas con recubrimiento reflectante en una de sus caras las cuales están enfrentadas paralelamente y muy próximas entre sí (5). La selección de la

¹Se llama así, a la medición del brillo (16). Y literalmente significa medición de luz

longitud de onda, en el centro de la imagen, que se transmite se consigue con variaciones muy pequeñas y precisas de la separación entre las láminas. Así pues, un filtro de este tipo puede asemejarse a un interferómetro de Fabry-Perot por lo que el perfil de estos filtros es una lorentziana y la transmitancia responde a la siguiente expresión.

$$T(\lambda) = \frac{1}{1 + \frac{4(\lambda - \lambda_0)^2}{\Delta\lambda_0^2}} \quad (2.1)$$

En la Ec. (2.1) $T(\lambda)$ es la cantidad de energía que el filtro permite pasar de la luz que le llega, λ y λ_0 son respectivamente, las longitudes de onda observada y en la que se centra un filtro sintonizable. Y $\Delta\lambda_0$ es la anchura de un filtro.

2.3. Origen de los datos: El GTC

El Gran Telescopio de Canarias (GTC), ver Fig. (2.1), es un telescopio reflector con un espejo primario de 10.4 metros de diámetro. Se encuentra situado en el Observatorio del Roque de los Muchachos del Instituto de Astrofísica de Canarias (IAC) en la isla canaria de La Palma (5). Su localización es un lugar cuya geografía y clima proporcionan unas condiciones favorables para la observación astronómica, estando a una altitud de 2400 metros sobre el nivel del mar (9).



Figura 2.1: *Fotografía del aspecto del GTC en El Roque de los Muchachos en La Palma, Canarias. IAC (9).*

Este telescopio tiene incorporados varios instrumentos científicos (9) entre los cuales cabe destacar OSIRIS.

2.4. OSIRIS

OSIRIS es el acrónimo de Optical System for Imaging and low-Intermediate-Resolution Integrated Spectroscopy. Se trata de un espectrógrafo para el rango de longitud de onda óptica, situado en el foco Nasmyth-B del GTC. Además de la imagen estándar de banda ancha y la espectroscopía, proporciona

capacidades adicionales, como la imagen de filtros sintonizables de banda estrecha. El hecho de que OSIRIS cuente con filtros sintonizables, en concreto dos, ya lo convierte en un instrumento puntero dentro de la observación astronómica, y, dota de carácter novedoso al trabajo que aquí se está tratando puesto que trabajos de este tipo se han venido realizando con filtros de bandas ancha y/o estrecha convencionales.

Los filtros sintonizables que dispone OSIRIS son: uno para el rango azul (BTF: λ 450 nm - 671 nm) y otro para el rango rojo (RTF: λ 651 nm - 934.5 nm)(5). El uso de los filtros sintonizables implica la utilización adicional de los filtros OS (*order sorter*) que evita la confusión entre los diferentes órdenes de interferencia del Fabry-Perot.

i	λ_0 (nm)	$\delta\lambda_0$ (nm)
1	736.54	1.37
2	738.54	1.36
3	739.55	1.36
4	740.54	1.36
5	741.55	1.35
6	742.54	1.35
7	744.53	1.35

Tabla 2.1: Las λ_0 son longitudes de onda centrales de cada filtro, y las $\delta\lambda_0$ son las anchuras de los mismos. La primera columna indica el número, i , con que cada filtro está enumerado. Por sencillez en la expresión, en el trabajo se hace referencia a cada una de las siete sintonizaciones del RTF como "filtros" sintonizables.

En la Tab. (2.1) se recogen las longitudes de onda centrales de los filtros con los que se han conseguido las imágenes astronómicas para este trabajo, junto con la anchura que abarca cada uno. Además de las siete imágenes de banda estrecha, obtenidas sintonizando el RTF, se tiene una octava con un filtro OS, que es la más profunda (9).

Además este instrumento cuenta con dos cámaras CCD. Se trata de dos Marconi CCD44-82 separadas un pequeño hueco entre las dos, y, que cubren un campo total, aproximadamente, de 7.8×7.8 " (10).

2.5. Base de datos: SDSS Sloan Digital Sky Survey

El Sloan Digital Sky Survey (SDSS), es un proyecto con el cual se consiguió hacer un mapa de una gran parte del cielo, y, del que cada cierto tiempo salen actualizaciones. Data Release 14 (DR14) es la tercera publicación de datos de la cuarta fase de SDSS (SDSS-IV). DR14 contiene observaciones de SDSS hasta julio de 2016.

El telescopio del SDSS con el que se hace el cartografiado se encuentra situado en el observatorio Apache Point de Nuevo México (5). La exploración cubre más de 7500 grados cuadrados con datos de casi 2 millones de objetos de los que 526000 son cuásares (siendo 144046 nuevos descubrimientos) (23).

Con todos los datos recopilados del cartografiado se creó un catálogo que contempla información de cada objeto, y, del que se van publicando varias versiones. El acceso a este catálogo es gratuito y

está disponible en Internet. Para este trabajo se ha usado la SDSS-IV(DR14) a la que se puede acceder a través de http://www.sdss.org/dr14/data_access/.

A partir del servidor SkyServer (22) se accede a la información catalogada del SDSS. Y tanto las imágenes como los espectros están disponibles de manera que mediante interfaces del SkyServer muy simples de usar se puede obtener, por ejemplo, una imagen de cualquier región del cielo cubierta por el SDSS, simplemente dando sus coordenadas. Puede accederse a través de <http://skyserver.sdss.org/dr14/en/tools/chart/navi.aspx>. De esta forma se han conseguido las imágenes de las Fig. (2.2) y Fig. (2.4) y los espectros de interés para este trabajo.

2.5.1. El cuásar.

El cuásar que ha suscitado el interés para la realización de la investigación de la que forma parte este trabajo, y con el que éste trabaja, es el SDSS J075618.13+410408.6 (ascensión recta 7h 56m 18.14s y declinación $41^{\circ} 4' 8.6''$). La razón de que éste cuásar haya sido elegido es que se detectó un exceso de fuentes submilimétricas en torno a él en la literatura (21).

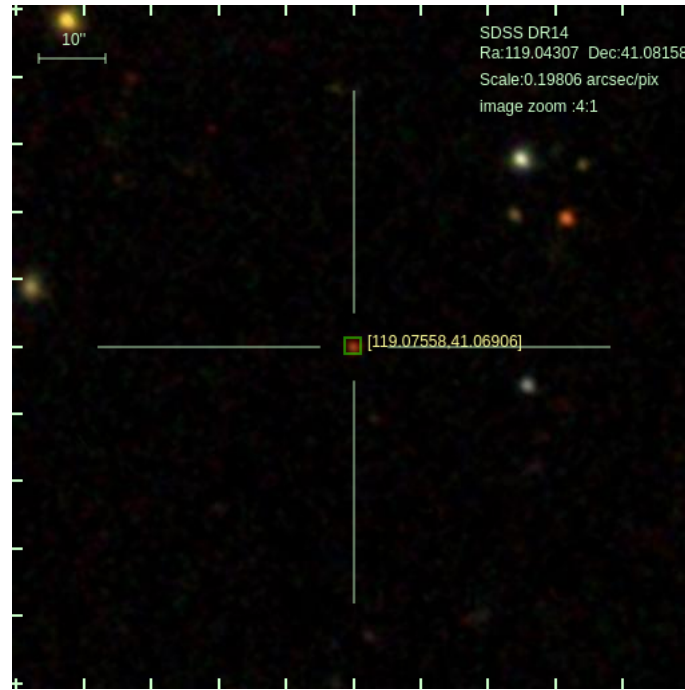


Figura 2.2: *Finding chart* realizada con SkyServer. El punto rojo justo en el centro de la imagen es el cuásar SDSS J075618.13+410408.6.

Mediante el uso de Skyserver hemos obtenido la *finding chart*² de la Fig. (2.2) y el espectro del cuásar mostrado en Fig. (2.3).

²En el ámbito de la astrofísica es así como se denomina a una imagen de posición del cielo.

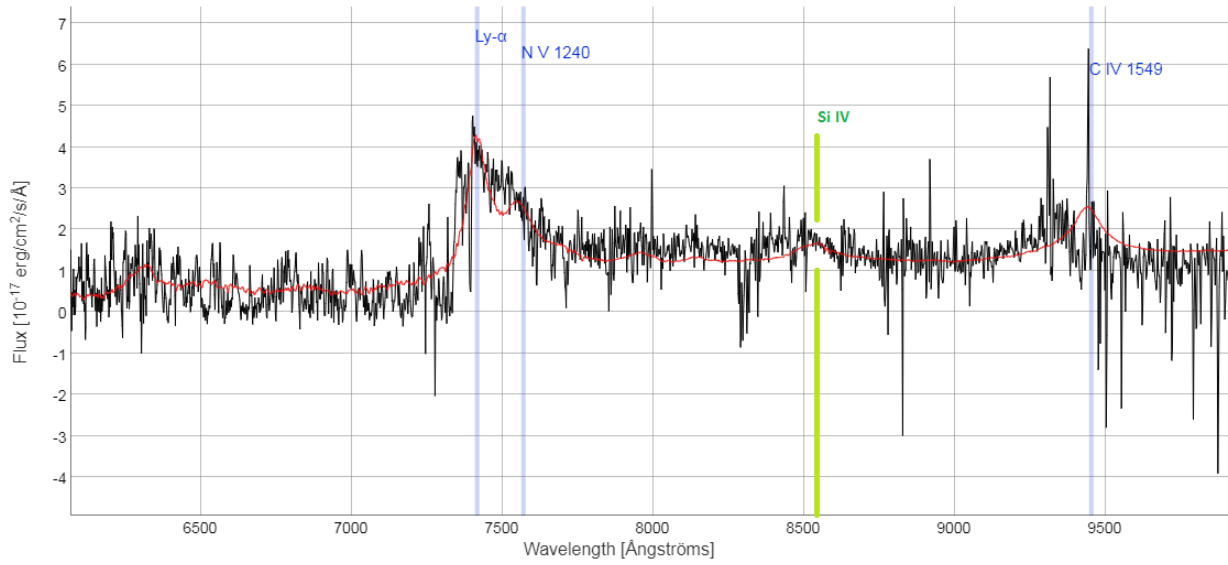


Figura 2.3: Espectro real, en negro, y modelo teórico del espectro, en rojo, del cuásar SDSS J075618.13+410408.6, realizado por SDSS. Puede observarse que el modelo teórico calculado por SDSS está desplazado frente a los datos observados. (1).

Durante el transcurso de este trabajo haciendo una inspección visual del espectro del cuásar, se observó que el *redshift* dado de manera automática en el catálogo de SDSS, $z = 5.10$, es incorrecto. En la Fig. (2.2), se observa que el modelo teórico, en rojo, está desplazado respecto del espectro real, en negro, lo que provoca que el z proporcionado por el SDSS no sea el que debería ser.

2.5.2. La estrella de calibración.

Como se explica posteriormente en la sección 3.2, para la calibración de las imágenes ha sido necesario usar una estrella.

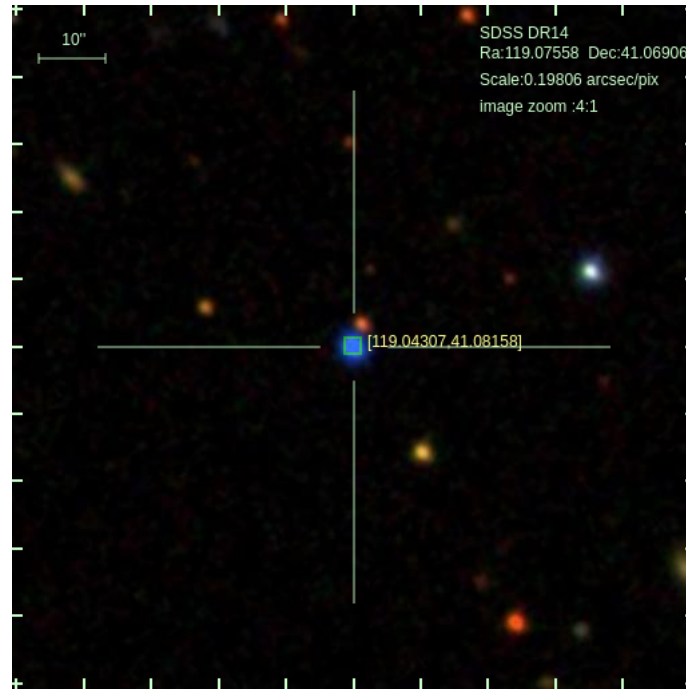


Figura 2.4: *Finding chart* de la estrella SDSS J075610.33+410453.6 usada en este trabajo para calibrar los flujos de todas las fuentes detectadas. Se ve en el centro la estrella como un punto azul.

La estrella que se ha escogido es la SDSS J075610.33+410453.6 (ascensión recta 7h 56m 10.34s y declinación 41° 4' 53.69") y puede verse su *finding chart* en la Fig. (2.4). Como se ve es una estrella de color azul y se trata de una estrella de tipo sdB, una enana blanca (25).

Capítulo 3

Metodología.

El procedimiento experimental seguido consiste en el desarrollo de un método para la selección de fuentes que sean candidatas a estar asociadas espacialmente al cuásar descrito en la sección previa. Para ello se ha tenido que ir extrayendo información de las ocho imágenes astronómicas, que son los únicos datos experimentales con los que la autora de este trabajo ha contado. Sin la posibilidad de poder hacer repeticiones de las medidas en caso de que se deseara, se ha tenido que ir buscando soluciones a los problemas que han ido surgiendo y que en esta sección se explican.

El método de selección creado ha conllevado un procedimiento que queda enmarcado en tres grandes fases. La primera fase consiste en la detección de fuentes y la extracción de información a partir de las imágenes astronómicas. La segunda fase consiste en la fotometría y el análisis de la información extraída. Y finalmente se llegará así a una tercera fase en la que se está en condiciones de crear un método de selección, aplicarlo y analizar los resultados que éste proporciona.

En esta sección se exponen la primera y segunda fase. A continuación se explica de forma conjunta el procedimiento seguido por la autora de este trabajo, en las que ambas quedan enmarcadas para llegar a la tercera fase que se corresponde con la sección 4.

En lo que sigue, se designará con PR, a toda tarea que haya sido realizada con uno de los programas o scripts creados para la realización de este trabajo. Una breve explicación de los mismos puede encontrarse en el apéndice B.

3.1. Concepto de flujo astronómico.

El flujo recibido desde una fuente en el contexto astronómico tiene su significado propio al margen del sentido que se le da en otros ámbitos. Así pues, en este contexto, se define flujo como la cantidad de energía recibida de un objeto astronómico que atraviesa un diferencial de área, dS , por unidad de tiempo e intervalo de longitud de onda $d\lambda$ (16).

$$F_{\lambda} = \frac{dE(\lambda)}{dt \, dS \, d\lambda} \quad (3.1)$$

siendo F_{λ} el flujo monocromático.

3.2. Fase 1: Detección de fuentes y extracción de datos de las imágenes astronómicas.

Para realizar la búsqueda de fuentes se dispone de una serie de imágenes, cada una referida a un filtro, que no están calibradas fotométricamente y que además hay pequeñas diferencias de posición entre ellas.

Antes de llegar a medir el flujo de las fuentes en unidades físicas, se tiene que localizar las fuentes y medir su flujo en cuentas. Después, en este trabajo se han asociado las fuentes en cada imagen entre ellas, para lo cual hay que basarse en su posición y posteriormente se han convertido esos flujos en cuentas a flujos físicos. Y entonces ya se consigue estar en condiciones de seleccionar las galaxias que se buscan. En lo que sigue se explicará en detalle cada uno de los pasos seguidos que forman cada fase del trabajo para realizar tal búsqueda:

Para encontrar fuentes y medir sus flujos usamos un programa estándar en astronomía que se llama SExtractor (2). Este programa proporciona las posiciones de las fuentes en coordenadas de cielo, el flujo en cuentas y el error en el flujo, que es lo necesario para este trabajo.

No conviene usar SExtractor con cada una de las ocho imágenes por separado y luego unir las porque como se espera que en nuestras fuentes de interés haya una caída apreciable de flujo hacia los filtros azules, puede ser que en alguno de los filtros azules una fuente en concreto no esté detectada. Sin embargo, necesitamos tener una medida del flujo en la posición de la fuente en los filtros más rojos. No obstante, esto no supone ningún problema, ya que SExtractor se puede correr con dos imágenes (PRF1). La primera imagen, denominada imagen de referencia, la usa para encontrar las fuentes, y la segunda para medir los flujos en las posiciones de las fuentes encontradas en la primera. Entonces, esto simplifica la labor a realizar porque de esta manera se consiguen flujos en las posiciones de las fuentes que interesan. Pero SExtractor tiene una limitación y es que la imagen de referencia y la imagen en la que se miden los flujos, tienen que tener el mismo tamaño y exactamente las mismas coordenadas. Entonces, lo que se ha hecho es en primer lugar escoger la imagen OS, como imagen de referencia, porque se ha comprobado que no hay ninguna fuente de las imágenes individuales, el resto de imágenes, que no esté en ella. Esta comprobación se ha realizado variando los parámetros de SExtractor y encontrando así la mejor combinación entre ellos (PRF1).

El conjunto de parámetros claves para la detección de fuentes en la imagen OS y los valores que se les ha asignado son:

- DETECT_THRESH = 2.
Establece el mínimo número de sigmas sobre el fondo para que un pixel sea considerado potencialmente correspondiente a una fuente.
- DETECT_MINAREA = 16.
Fija el mínimo número de píxeles adyacentes que han pasado el filtro anterior para considerar que se ha detectado una fuente.
- DEBLEND_MINCONT = 0.01.
Parámetro de contraste para la situaciones de haber dos fuentes muy próximas entre sí. Indica el valor a partir del cual, SExtractor separa un conjunto de píxeles significativos adyacentes en más de una fuente.

• PHOT.APERTURES = 4.

Parámetro con el que se varía el diámetro del círculo de extracción que usa Sextractor, en píxeles (cada píxel corresponde a $0.125''$).

Una inspección previa de las imágenes a simple vista reveló que hay una serie de artefactos, defectos en las imágenes en las partes externas. Esto ha conllevado a que se tenga que buscar y definir una region buena y común (PR2) a todas las imágenes astronómicas originales, en coordenadas de cielo y que no presente ninguna de estas imperfecciones. Entonces las apreciaciones que se hagan a partir de ahora sobre el número de fuentes y su calidad se van a referir a esta zona común de buena calidad.

Con el conjunto de parámetros fijado, se consigue que Sextractor por un lado, no deje de detectar las fuentes que se pueden ver a ojo y por otro lado, no detecte fuentes espurias. Después se detectan las fuentes en la imagen OS y se generan unas imágenes, que aquí serán llamadas imágenes simuladas, con las mismas coordenadas de cada uno de los filtros pero colocando las fuentes artificiales en las mismas posiciones que en la imagen de referencia, (PR0 y PR1).

Seguidamente se ha corrido Sextractor siete veces en el modo de dos imágenes, usando como referencia para cada filtro la imagen simulada correspondiente. Con esto se han conseguido siete catálogos (PRF1), cada cual correspondiente a un filtro. El siguiente paso ha sido extraer la información de interés (ascensión recta, declinación, flujos en cuentas y errores). De esta forma se han conseguido nuevos catálogos (PR3), pero referidos solo a la región común libre de artefactos y en un formato manipulable por los programas con los que se han tratado con posterioridad.

Seguidamente se han correlado estos catálogos, mediante un programa común en astronomía llamado Topcat (15), consiguiendo así, (PR4), un único fichero con la información de todas las fuentes detectadas en todos los filtros. En este catálogo los flujos están en unidades de cuentas. Y para finalizar la primera fase, a partir de este último fichero obtenido se escriben las instrucciones en el formato adecuado para crear las *finding charts* haciendo uso de una aplicación (PR5) de visualización de datos e imágenes astronómicas llamada SAOImage DS9.

3.3. Fase 2: Calibración fotométrica y análisis de la información.

En el archivo obtenido, al finalizar la primera fase, ya se disponen todos los flujos y cuentas, junto con sus respectivos errores, (además también alberga las coordenadas de ascensión recta y declinación de cada fuente) de todas las fuentes entre las que posteriormente se hará la selección de las candidatas a cumplir los requisitos físicos para cumplir el objetivo deseado.

Así pues, llegado este punto, se procede a la calibración fotométrica de las fuentes (PR6) del citado archivo. Para hacerlo se busca una fuente que esté en todas las imágenes cuyo flujo sea conocido. Y por medio de SDSS se disponen de varias pero en concreto para este trabajo se ha escogido la estrella presentada en la sección 2.5.2, y que, desde ahora en adelante se hace alusión a ella denominándola fuente de calibración. Del catálogo de SDSS se descarga el espectro de esta estrella. Los datos SDSS que describen las longitudes de onda de la línea espectral utilizan longitudes de onda de vacío. Sin embargo, las longitudes de onda de las transiciones atómicas generalmente se indican a temperatura y

presión estándar (aire).

$$\lambda_{\text{Aire}} = \lambda_{\text{Vac}} / (1.0 + 2.735182 \cdot 10^{-4} + 131.4182 / \lambda_{\text{Vac}}^2 + 2.762498 / \lambda_{\text{Vac}}^4) \quad (3.2)$$

Por ello lo primero que se ha hecho con el espectro es hacer la conversión de vacío a aire por medio de la Ec. (3.2), en la cual λ_{Vac} es la longitud de onda de vacío en ángstroms, y λ_{Aire} es la longitud de onda de aire. Esta expresión es la conversión estándar (IAU ¹) de longitudes de onda de vacío a aire (18).

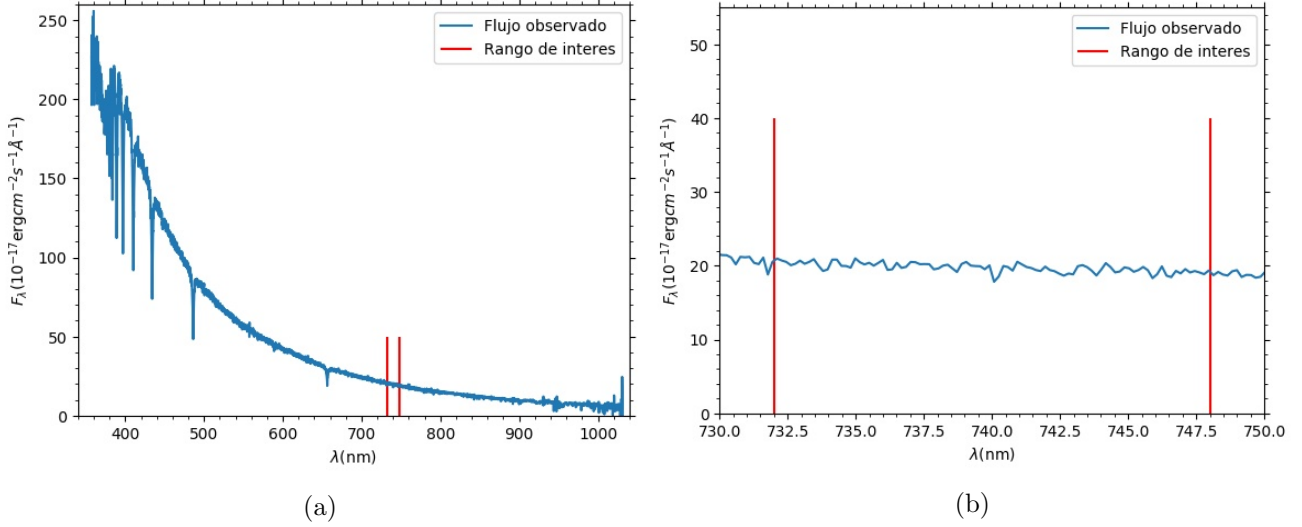


Figura 3.1: *Espectro de la estrella de calibración. Las líneas rojas indican el rango que abarcan los filtros empleados. En (a) se muestra el espectro completo y en (b) una ampliación de la zona abarcada por los filtros. (PR6).*

La elección de esta fuente para realizar la calibración se debe a que en la zona donde están los filtros tiene un espectro sin líneas de absorción ni emisión con una variación suave, ver Fig. (3.1.a), a diferencia de lo que ocurre en el mismo rango en el espectro del cuásar. En la Fig.(3.1.b) se muestra ampliado el rango del espectro que abarcan en su conjunto los filtros empleados. Puede observarse como es una zona del espectro razonablemente plana y que presenta un buen cociente señal/ruido.

Lo que se observa en las imágenes astronómicas son cuentas y lo que se necesita son flujos físicos. Entonces se ha calculado una constante de proporcionalidad que permita hacer la conversión entre cuentas y flujos físicos.

Para llevar a cabo esto se han aplicado las siguientes expresiones.

$$F_i = \frac{\int F_\lambda(\lambda) T_i(\lambda) d\lambda}{\int T_i(\lambda) d\lambda} \quad (3.3)$$

donde F_i es el flujo observado en unidades físicas a través del filtro i , y T_i es la transmitancia del filtro i a partir de la Ec.(2.1) y la Tab.(2.1). Y $F_\lambda(\lambda)$ es el flujo antes de atravesar la atmósfera.

$$F_i = C_i O_i \quad (3.4)$$

¹International Astronomical Union.

donde O_i es el flujo observado en cuentas y C_i es la constante de calibración.

Se tiene la ventaja de que para la fuente de calibración además de tener los flujos en cuentas, C_i , se tienen los flujos físicos, F_i , para cada filtro. Entonces, por medio de la Ec.(3.4) se obtiene la constante de calibración para cada filtro. Cada constante de calibración se multiplica por todos los flujos obtenidos del filtro asociado a cada una. De esta forma todos los flujos en cuentas quedan reescalados a flujos físicos. Para comprobar la calibración se superponen los flujos calibrados asociados a cada filtro sobre el flujo observado de la estrella.

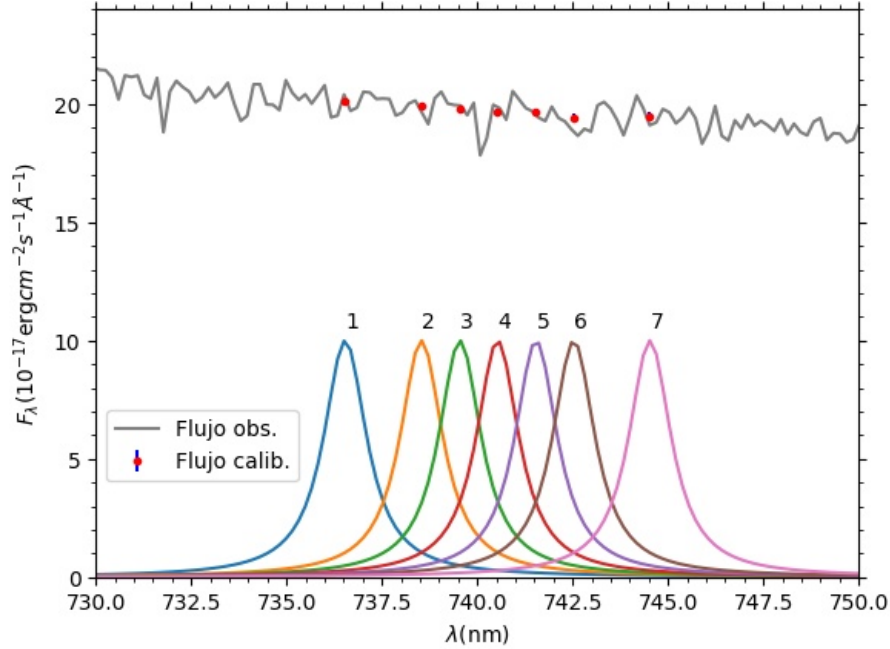


Figura 3.2: *Flujo observado en unidades físicas F_λ de la estrella de calibración junto con el flujo calibrado que ha atravesado su respectivo filtro. En distintos colores y enumerados, según la Tab. (2.1), se representan los perfiles lorentzianos de cada filtro, que han sido multiplicados por un factor 10^7 para que se vean mejor. Sus formas se corresponden con la Ec. (2.1). (PR6).*

A la vista de la Fig. (3.2) puede observarse que efectivamente, los flujos calibrados están sobre la línea gris, los flujos físicos de la estrella. Queda reflejado así la correcta calibración de los flujos en cuentas a magnitudes físicas.

Llegado este punto ya se dispone de una lista de fuentes con sus flujos calibrados. Con ello se diseña un programa de selección de galaxias LABs y LAEs que es lo que se presenta en la sección de resultados.

3.4. Descentrado de los filtros.

Se ha trabajado con siete filtros porque las galaxias se mueven (por la ley de Kepler) en el pozo de potencial del cúmulo. Y es que no todas las galaxias, debido a sus movimientos propios, van a estar al mismo *redshift* que el cuásar. Precisamente por esto, se pidieron siete sintonizaciones del filtro, al z

del cuáasar (determinado incorrectamente) y tres a cada lado (hacia el azul y hacia el rojo), cubriendo 1000 km s^{-1} por ser ésta la velocidad típica de recesión de un cuáasar (8) (presumiblemente situado en el centro del cúmulo) ya que se espera que las dispersiones en los movimientos propios de las galaxias en el cúmulo hagan que sus líneas de emisión Lyman α estén corridas hacia el rojo y el azul de la posición de esa línea para el cuáasar.

Sin embargo, durante la realización de este trabajo la autora del mismo observó que el modelo del espectro teórico del SDSS para el cuáasar está desplazado respecto de su espectro observado. A partir de esto, detectó que el problema está en que el *redshift* proporcionado por SDSS, $z = 5.10$, es incorrecto. Y este valor del *redshift* es el que se dió en la propuesta (26) de la investigación de la que surge este trabajo, y con el que ésta fue diseñada. Con lo cual las observaciones se hicieron con los filtros descentrados frente a donde deberían estar. Esto, debido a que no se puede repetir las observaciones, ha conllevado a improvisar y afrontar el imprevisto no como un problema sino como parte del aprendizaje del tipo de problemas con los que uno se puede encontrar en el ámbito de la investigación. Además, las medidas son válidas siempre y cuando se tenga en cuenta el descentrado. Entonces lo que se ha hecho es medir el *redshift* correcto y calcular para este nuevo valor con qué longitud de onda se corresponde la línea Lyman- α para las galaxias que estén en torno al cuáasar. Y tal línea ya no estará centrada en el filtro 4, tal como se pretendía inicialmente.

3.4.1. Midiendo el *redshift*.

Para medir el redshift de la fuente se ha hecho un ajuste del espectro del cuáasar de las líneas de litio SiIV, (1397 Å), y carbono CIV, (1548 Å). No se ha empleado la línea Lyman- α porque debido a la absorción, falta la mitad de la línea. Se ha utilizado un modelo que es una línea recta (para el continuo) y dos gaussianas, una por cada línea, donde los centros de las gaussianas se les ha constreñido que conserven la relación entre las longitudes de onda esperadas y a sus respectivas anchuras se les ha obligado que sean las mismas.

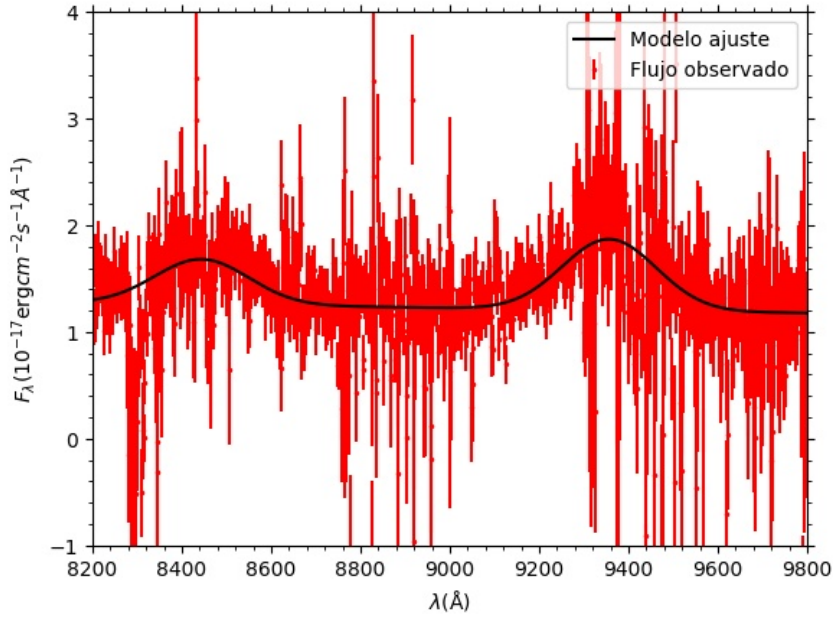


Figura 3.3: Ajuste χ^2 del flujo observado, F_λ , del espectro del cuáasar extraído de SDSS. En negro el modelo de ajuste y en rojo el espectro observado. (PR8)

El ajuste se ha hecho minimizando χ^2 obteniendo así un mejor ajuste, ver Fig. (3.3). De esta forma se ha obtenido con bastante precisión para el SiIV la longitud de onda $\lambda_{\text{SiIV}} = 8444.3 \text{ \AA}$. Se ha usado esta línea porque en una inspección visual previa, se ha observado que la posición donde está es bastante limpia (sin absorción).

$$z = \frac{\lambda_s}{\lambda_0} - 1 \quad (3.5)$$

Se ha empleado la Ec.(3.5) en la cual λ_s es la longitud estimada medida en el espectro observado del cuáasar y λ_0 la longitud de onda en reposo de la línea en cuestión. Entonces, sustituyendo la longitud de onda del SiIV obtenida del ajuste de la Fig. (3.3) y la que tiene en el reposo, el resultado obtenido para el *redshift* del cuáasar es $z = 5.04$.

3.5. Forma de obtener resultados con los datos disponibles.

El error en el z ha hecho que las longitudes de onda centrales de los filtros ya no están distribuidas alrededor de la posición esperada de la línea Lyman alfa del cuáasar, y que estén desplazados hacia el rojo respecto de ella. Por tanto, se ha trabajado con una configuración de filtros en la que ya no hay tres a cada lado respecto de donde se encuentra el cuáasar. Y por ello lo que se ha podido hacer es buscar las galaxias que tienen un movimiento propio corrido al rojo con respecto al cuáasar, lo que implica velocidades propias positivas.

Entonces, lo que se ha hecho para el z correcto, es estimar la longitud de onda que se corresponde con la emisión de Lyman- α , considerando una velocidad típica de recesión de las galaxias en el

cúmulo de $+1000 \text{ km s}^{-1}$ (8). Y se ha añadido esta velocidad a la correspondiente velocidad de recesión del cuásar obteniéndose que la línea Lyman- α se produce a la longitud de onda de 737.45 nm, que está situada entre los filtros 1 y 2.

Por tanto, gracias a la configuración de filtros inicialmente establecida, pese al error en z , se ha podido recuperar parte de los objetivos científicos que se querían abordar, ya que los dos filtros más azules de los siete (el 1 y el 2) están a ambos lados de la longitud de onda a la que se espera la emisión de Lyman- α de una galaxia que tenga una velocidad de recesión típica.

Esto entonces, permite seleccionar las galaxias LAB con esa velocidad propia, como aquellas en las que el flujo en el filtro 1 es mucho menor que en el filtro 2 y siguientes. Similarmente, las galaxias que presenten un exceso de emisión en el filtro 2 frente a las siguientes serían candidatas a ser LAE con esa velocidad propia.

En definitiva, en este trabajo las galaxias que se han podido analizar son aquellas que se alejan de nosotros a 1000 km s^{-1} con respecto al cuásar.

3.6. Señal/ruido

En los experimentos con cuenta de fotones, cuando el número de cuentas es muy alto ha de usarse la estadística de Poisson. Se tiene así que la señal/ruido.

$$\left. \begin{array}{l} S + B = T \\ B \end{array} \right\} \begin{array}{l} \text{señal} = T - B \\ \text{ruido} = \sqrt{T + B} \end{array}$$

Cociente señal-ruido:

$$\text{SNR} = \frac{\text{señal}}{\text{ruido}} = \frac{T - B}{\sqrt{T + B}} \propto \frac{t}{\sqrt{t}}$$

En las expresiones anteriores S es el número de cuentas en la fuente, B es el número de cuentas en el fondo y T es la suma de $S + B$ que indica el total de cuentas que se mide en la zona de la fuente (4). Si tanto las cuentas de la fuente S como en el fondo B son proporcionales al tiempo de exposición están relacionados tal como sigue,

$$\text{SNR} \propto \sqrt{t} \quad (3.6)$$

De esta expresión puede inferirse que cuanto mayor sea el tiempo de exposición, dado que la señal se puede suponer que no cambia en escalas temporales del orden del tiempo de exposición, mayor será el cociente señal/ruido. Pero también se deduce que hay que elevar mucho el tiempo de exposición para notar cambios significativos, ya que este aumenta con la raíz cuadrada.

Capítulo 4

Resultados

Los resultados de este trabajo son por una parte el método de selección creado en base al tipo y formato de datos tratados, y, por otro lado la clasificación de objetos realizada con dicho método con el fundamento físico explicado en la sección 1.3. Los resultados de esta sección se complementan con las tablas aportadas en el apéndice B.

4.1. Fase 3: Método de selección y su descripción física.

El método que aquí se presenta realiza una selección del total de fuentes detectadas pero que no contempla los filtros 5, 6, y, 7, ver Fig. (3.2). Se trata de un método en primera aproximación que consiste en establecer una serie de criterios matemáticos con fundamento físico, de cara a establecer una clasificación de los objetos que se quieren encontrar (PR8). Estos criterios matemáticos, el método los evalúa sobre los flujos calibrados con las 986 fuentes candidatas. Así pues, según las condiciones que la fuente cumple, el método la clasifica como fuente interesante (ver más adelante). Si no cumple ninguna condición será considerada como fuente no interesante para el objetivo principal de este trabajo.

Con este método se hace una selección entre 986 fuentes encontradas en este trabajo y se encuentran 63 candidatas. Evaluando el método con distintos conjuntos de parámetros se ha reducido el número de candidatas y se llega a que las más favorables son un total de 4.

El método se ha diseñado de forma que busque objetos que se aproximen dentro de la incertidumbre de las medidas, a formas esperadas de los espectros de las galaxias en torno al cuásar en las que centramos la búsqueda. Estos modelos son explicados en forma de unos diagramas creados (PR10) por la autora, ver Fig. (4.1).

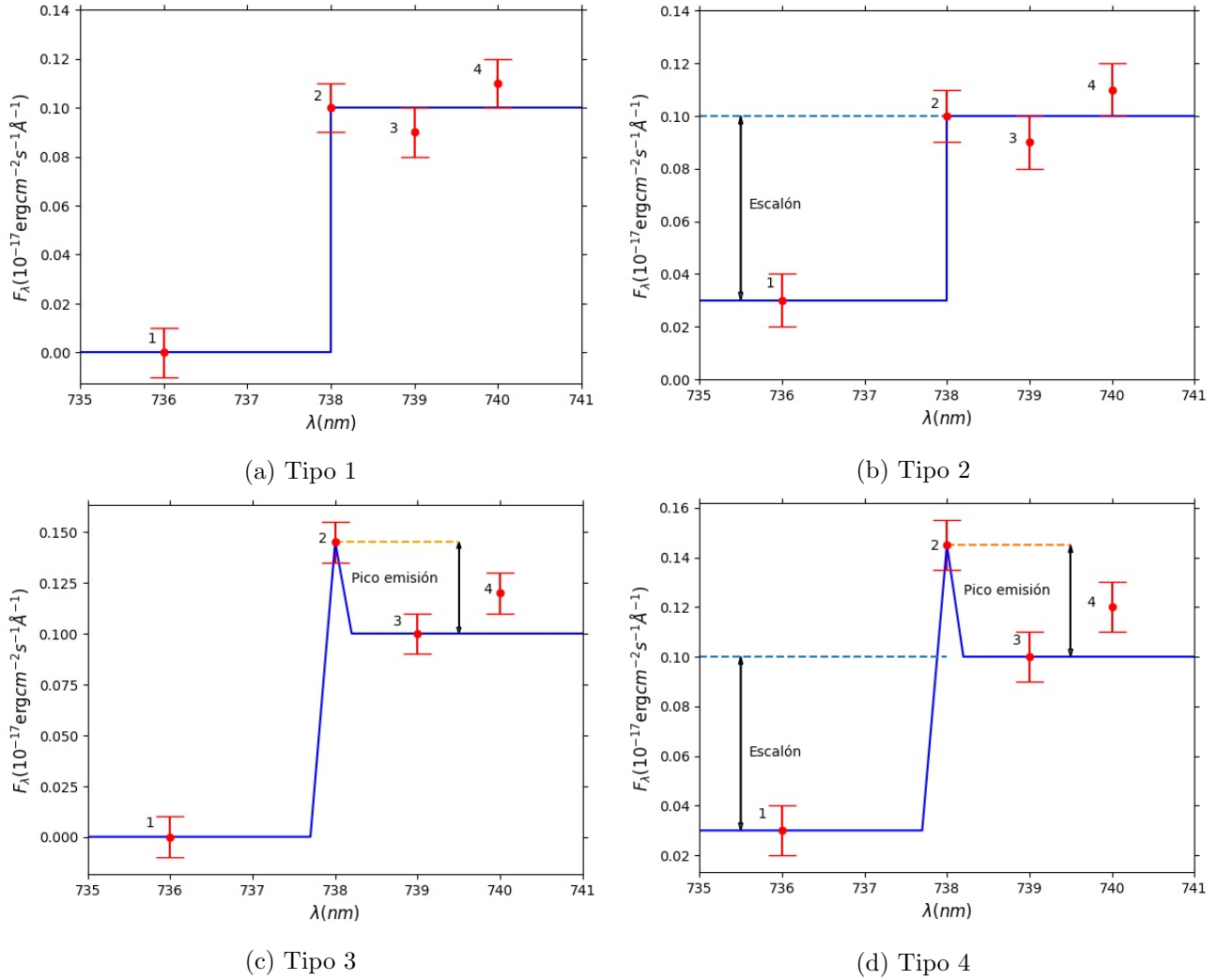


Figura 4.1: Diagramas ideales de los de los diferentes tipos en que clasifica el método de selección a las galaxias que selecciona como candidatas. (PR11).

Si se observa la Fig. (4.1) se ve que el escalón, va a estar presente siempre, debido a la absorción de radiación por el hidrógeno intergaláctico. Pero en los llamados tipo 1 y tipo 3, dado que en ellos el filtro 1 no detecta nada, el escalón no se puede medir y por ello no aparece dibujado. Por contra dicho escalón en el tipo 2 y 4 dado que en estos casos el filtro 1 detecta señal, se puede medir.

Una característica común entre los de tipo 1 y tipo 2 es que el filtro 2 y el filtro 3 detectan flujos similares. Sin embargo esta continuidad no existe en los casos de tipo 3 y tipo 4. En ellos es característica la presencia de un *pico de emisión*, esto es lo que se corresponde con la línea Lyman- α . Luego para que una galaxia sea clasificada como tipo 3 o tipo 4, su espectro no solo debe presentar un salto brusco entre el filtro 1 y el filtro 3, si no también ha de presentar un exceso de emisión entre el filtro 2 y el filtro 3.

En base a la teoría explicada en la sección 1.4, el salto brusco está presente en los cuatro tipos y así se ha impuesto en el método.

La descripción de cada uno de estos diagramas se resume con las expresiones que definen cualitativa-

mete los criterios físicos en los cuales se basa el método. Estas son las siguientes:

$$\begin{array}{ll} \text{Tipo 1:} \left\{ \begin{array}{l} F_1 \text{ no detectado} \\ F_2, F_3, F_4 \text{ detectado} \\ F_2 \approx F_3 \\ F_3 \approx F_4 \\ - \end{array} \right. & \text{Tipo 2:} \left\{ \begin{array}{l} F_1 \text{ detectado} \\ F_2, F_3, F_4 \text{ detectado} \\ F_2 \approx F_3 \\ F_3 \approx F_4 \\ \frac{F_2}{F_1}, \text{ escalón como el del cuásar} \end{array} \right. \end{array}$$

$$\begin{array}{ll} \text{Tipo 3:} \left\{ \begin{array}{l} F_1 \text{ no detectado} \\ F_2, F_3, F_4 \text{ detectado} \\ F_2 > F_3 \text{ Línea significativa} \\ F_3 \approx F_4 \\ - \end{array} \right. & \text{Tipo 4:} \left\{ \begin{array}{l} F_1 \text{ detectado} \\ F_2, F_3, F_4 \text{ detectado} \\ F_2 > F_3 \text{ Línea significativa} \\ F_3 \approx F_4 \\ \frac{F_3}{F_1}, \text{ escalón como el del cuásar} \end{array} \right. \end{array}$$

F_1 , F_2 , F_3 y F_4 se refieren a el flujo de los filtros 1, 2, 3 y 4 respectivamente. Los tipos 1 y 2 se refieren a galaxias LAB y los tipos 3 y 4 a galaxias LAE.

Estos criterios físicos se traducen a términos matemáticos dando lugar a las condiciones matemáticas con las que se crea el método y a partir de las cuales la galaxia es de un tipo u otro. Estas son las que siguen:

$$\frac{F_2 - F_3}{\sqrt{(\Delta F_2^2 + \Delta F_3^2)}} \equiv \alpha_{2,3} > \alpha_{lim} \quad (4.1)$$

Donde $\alpha_{2,3}$ hace referencia a la línea de emisión entre el flujo detectado por el filtro 2 y el detectado por el filtro 3. Siendo α_{lim} el valor límite de α por encima del cual en este trabajo se considera que se detecta una línea de emisión.

$$\begin{array}{l} F_i \approx F_j \\ i, j = 2, 3 \quad i, j = 3, 4 \end{array} \Rightarrow \frac{|F_i - F_j|}{\sqrt{(\Delta F_i^2 + \Delta F_j^2)}} \equiv \beta_{i,j} \leq \beta_{lim} \quad (4.2)$$

Donde $\beta_{i,j}$, se refiere a la parte continua que se busca hacia el rojo del escalón. Y la inecuación se interpreta como cuan grande puede ser la diferencia entre la detección del filtro 3 y el filtro 2 así como del filtro 3 y el filtro 4, para que se considere que detectan la misma cantidad de radiación. Siendo β_{lim} el valor de $\beta_{i,j}$ para el cual o por debajo del cual, se considera que el continuo es constante.

$$\begin{array}{l} X = \frac{F_i}{F_j} \\ i, j = 1, 2 \quad i, j = 1, 3 \end{array} \Rightarrow \frac{|X - A|}{\Delta X} \equiv \gamma_{i,j} \leq \gamma_{lim} \quad (4.3)$$

Siendo A el cociente observado (en el cuásar) de flujos a un lado y otro de la línea Lyman y significa la fracción de flujo que se ha perdido en el continuo por absorción en el medio intergaláctico. La absorción de flujo es evaluada, respecto al flujo medido por el filtro 1, tanto para el flujo medido por el filtro 2 (tipo 2), como el flujo medido por el filtro 3 (tipo 4).

La condición de la Ec. (4.3) establece que el salto medido tenga un valor similar al medido en el cuásar, dentro de las incertidumbres. Saltos significativamente distintos, implicarían recorridos de la radiación en el medio intergaláctico incompatibles con el del cuásar.

El valor de A se obtiene a partir del espectro observado del cuásar. Se ha buscado unos rangos de longitudes de onda en el espectro del modelo que aparentemente se corresponden con una zona plana libre de líneas de absorción y emisión. Pero dado que el espectro del flujo observado del cuásar generado por SDSS presenta mucho ruido, se ha creado, (PR7), de nuevo el espectro observado pero suavizado con un filtro de anchura 1.36 nm y se ha dibujado sobre el espectro observado del cuásar, ver Fig. (4.2). Para los rangos de longitudes de onda estimados se han calculado las correspondientes longitudes de onda en el espectro observado. Para la zona hacia el rojo el rango escogido ha sido (815, 830) nm. Al llevar las longitudes de onda extremas de este rango a la Fig. (4.2) se ha encontrado que no se puede llegar hasta el valor 830 nm porque ahí empieza una línea de absorción. Por esto, para esta zona se ha tenido que reducir el rango. Con lo cual en la zona hacia el rojo el rango escogido es (818.71, 825.81) nm. Y se ha calculado el promedio de los flujos suavizados en ese rango, ver Fig. (4.2). De esta forma se ha conseguido un flujo promedio para la zona hacia el rojo de aproximadamente $1.4 \times 10^{-17} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1}$, ver Fig. (4.2).

Análogamente se ha obtenido el flujo en la zona hacia el azul. Sin embargo en esta zona el rango estimado en SDSS y traducido a longitudes de onda observadas, ha sido (686.15, 694.24) nm y al llevarlo sobre el espectro observado se ha visto que se corresponde con una zona aprovechable. Con lo cual se ha promediado todo este rango de longitudes de onda, y, con la longitud de onda promedio se ha extraído de la Fig. (4.2) el flujo promedio correspondiente, para la zona hacia el azul, el cual es aproximadamente $0.4 \times 10^{-17} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1}$, ver Fig. (4.2).

Ahora, en base a la definición dada para A se ha hecho el cociente de los flujos promedios obtenidos. De esta forma se ha obtenido la cantidad de flujo del cuásar absorbida, siendo $A = 3.5$.

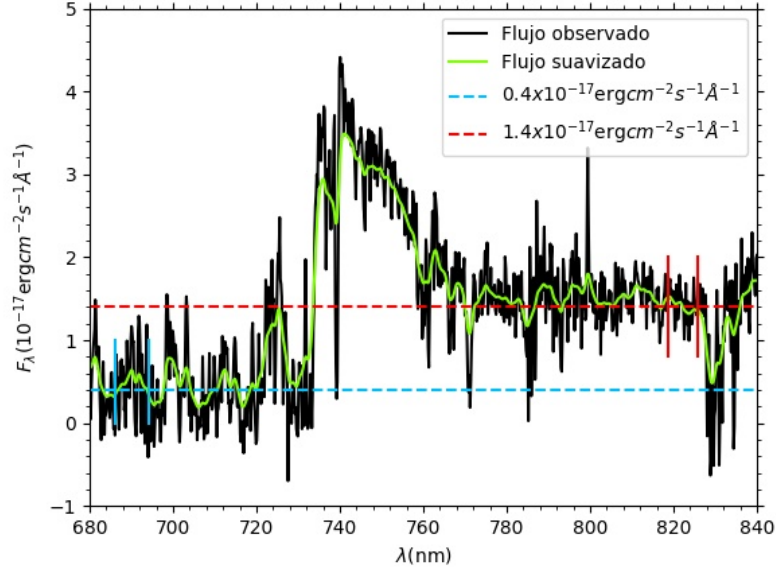


Figura 4.2: Representación gráfica del espectro observado suavizado con un filtro de 13.6 nm, en verde, sobre el flujo observado del cuásar, en negro. Las líneas discontinuas marcan los flujos promedios obtenidos, $1.4 \times 10^{-17} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ Å}^{-1}$ para la parte hacia el rojo, línea discontinua roja. Este flujo se corresponde con el promedio de los flujos en el rango de longitudes de ondas marcado con las líneas verticales rojas. Y $0.4 \times 10^{-17} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ Å}^{-1}$ hacia el azul, línea discontinua azul, el cual se corresponde con el promedio de flujos en el rango comprendido entre el par de líneas verticales azules. (PR7).

Para cada uno de los tipos de galaxias de la clasificación hecha en este trabajo, las condiciones en términos matemáticos están asociadas a cada uno del siguiente modo:

$$\text{Tipo 1: } \begin{cases} F_1 \leq 0 \\ F_2, F_3, F_4 > 0 \\ \beta_{2,3} \leq \beta_{lim} \\ \beta_{3,4} \leq \beta_{lim} \\ - \end{cases} \quad \text{Tipo 2: } \begin{cases} F_1 > 0 \\ F_2, F_3, F_4 > 0 \\ \beta_{2,3} \leq \beta_{lim} \\ \beta_{3,4} \leq \beta_{lim} \\ \gamma_{1,2} \leq \gamma_{lim} \end{cases}$$

$$\text{Tipo 3: } \begin{cases} F_1 \leq 0 \\ F_2, F_3, F_4 > 0 \\ \alpha > \alpha_{lim} \\ \beta_{3,4} \leq \beta_{lim} \\ - \end{cases} \quad \text{Tipo 4: } \begin{cases} F_1 > 0 \\ F_2, F_3, F_4 > 0 \\ \alpha > \alpha_{lim} \\ \beta_{3,4} \leq \beta_{lim} \\ \gamma_{1,3} \leq \gamma_{lim} \end{cases}$$

4.2. Selección de posibles candidatas.

Al aplicar el método con diferentes conjuntos de parámetros con las condiciones matemáticas fijadas, se consiguen los siguientes resultados en cuanto al número de galaxias encontradas en función de dichos parámetros.

Casos	γ_{lim}	β_{lim}	α_{lim}	Tipo 1	Tipo 2	Tipo 3	Tipo 4
A	2	1	2	6	56	0	1
B	1	1	2	6	31	0	1
C	0.5	1	2	6	21	0	1
D	2	0.5	2	3	15	0	0
E	1	0.5	2	3	10	0	0
F	0.5	0.5	2	3	7	0	0
G	2	0.25	2	2	5	0	0
H	1	0.25	2	2	3	0	0
I	0.5	0.25	2	2	2	0	0

Tabla 4.1: *Resultados del número de galaxias candidatas de estar en torno al cuásar, al aplicar a las 986 fuentes detectadas el método de selección. Donde α_{lim} , β_{lim} y γ_{lim} son los valores límite fijados para las diferencias divididas por los errores comunes, correspondiéndose respectivamente las ecuaciones. Ec. (4.1), Ec. (4.2) y Ec. (4.3).*

El caso A de la Tab. (4.1), es la muestra que contempla el total de fuentes seleccionadas. El resto de casos de la tabla citada, son submuestras de dicho caso.

Los resultados de las ecuaciones Ec. (4.1), Ec. (4.2) y Ec. (4.3) para las 63 fuentes que en total se han seleccionado se aportan en las Tab. (6.1) y Tab. (6.2) del apéndice A. Además, tales resultados se han ilustrado con histogramas, ver Fig. (6.1), aportados también en el apéndice A.

Capítulo 5

Discusión

Resulta interesante realizar una discusión crítica de los resultados que se han obtenido a lo largo de este trabajo, prestando atención al método de selección. Fijarse en cómo funciona el método permite además encontrar posibles limitaciones. Se considerará el efecto de dichas limitaciones sobre los resultados de las fuentes detectadas, especialmente sobre el caso que engloba a las mejores candidatas de entre todas, el caso I de la Tab. (4.1), proponiendo futuras líneas de investigación con las que hacer frente a ellas, y proseguir en el estudio de la investigación de la que forma parte este trabajo, la búsqueda de una sobredensidad en torno a un cuásar a alto *redshift*.

5.1. Limitaciones iniciales.

Durante el transcurso de este trabajo habido un problema siempre presente: el descentrado de los filtros sintonizables frente a la posición observada.

Y este problema, pese a no ser un problema generado en el algún paso de este trabajo, y en particular no ser causado por un error cometido en el método de selección creado, sí que afecta a éste directamente. Esto entonces ya obliga a usar el método en condiciones desfavorables.

El problema del descentrado de los filtros ha alterado a la idea que se tenía en el inicio de como buscar las galaxias. En un principio como en promedio la mayor parte de las galaxias van a tener una velocidad radial propia observada pequeña, se esperaba encontrar en sus espectros el escalón cerca de la longitud de onda observada de Lyman- α . Sin embargo en este trabajo como consecuencia del descentrado, ha sido obligado buscarlas con una velocidad radial propia alta y positiva (corrimiento hacia el rojo), lo que reduce mucho el número de galaxias que se esperaba encontrar. Con lo cual una primera limitación está en el número de candidatas que pueden encontrarse siendo este menor del que se esperaba.

5.2. Fuentes seleccionadas.

En la Tab. (4.1) se recogen los resultados. El número de fuentes seleccionadas disminuye conforme la lectura de la tabla se hace desde el caso A hacia el caso I. Esto es debido a que se van imponiendo condiciones cada vez más exigentes. Como estas condiciones se imponen sobre cantidades que son diferencias divididas por el error combinado, al final se favorecen las galaxias cuyos flujos tienen mayores barras de error. Sin embargo, en el caso I, se ha encontrado que hay candidatas visualmente mejores, lo cual se justifica en el siguiente apartado, que en el resto de casos. Pero cabe decir que hay un total de 63 fuentes (caso A) que cumplen las condiciones matemáticamente pero debido a las grandes barras

de error, no se puede asegurar con certeza que estén al mismo redshift que el cuásar.

5.3. Caso I: La mejor selección.

Se presta especial atención al caso I en el que los parámetros establecidos dan lugar a las mejores candidatas para encontrarse formando parte de una sobredensidad en torno al cuásar. Se trata de las mejores candidatas porque son las que además de cumplir las condiciones matemáticas, son las que cumplen la teoría física explicada en la sección 4. Esto se ve al comprobar que la distribución de sus flujos calibrados es acorde con los modelos teóricos de los tipos de galaxias explicados en la Fig. (4.1). Dentro de este caso, son en concreto dos de las fuentes las que mejor cumplen lo esperado. Esto ya supone un punto de partida para refinar la búsqueda realizada con este trabajo.

5.3.1. La mejor candidata: 303.

La mejor candidata es la fuente número 303 y se muestra en la Fig. (5.1). En la Fig. (5.1.a) se muestra su *finding chart*. Y se observa que es una fuente tan débil que a penas se puede apreciar en las imágenes salvo en la que se muestra en la imagen astronómica obtenida con el filtro OS, siendo en este donde se observa con claridad su presencia.

Fijándose ahora en el gráfico de la Fig. (5.1.b), se ve claramente el escalón entre la medida correspondiente del filtro 1 y la del filtro 2, característico de las galaxias de tipo 1. También, nótese la forma plana y continua, teniendo en cuenta las barras de error, que adoptan las medidas desde la medida del filtro 2 hasta la del filtro 4.

En el caso de esta fuente, las barras de error no son impedimento para garantizar que se trata de una buena candidata de tipo 1.

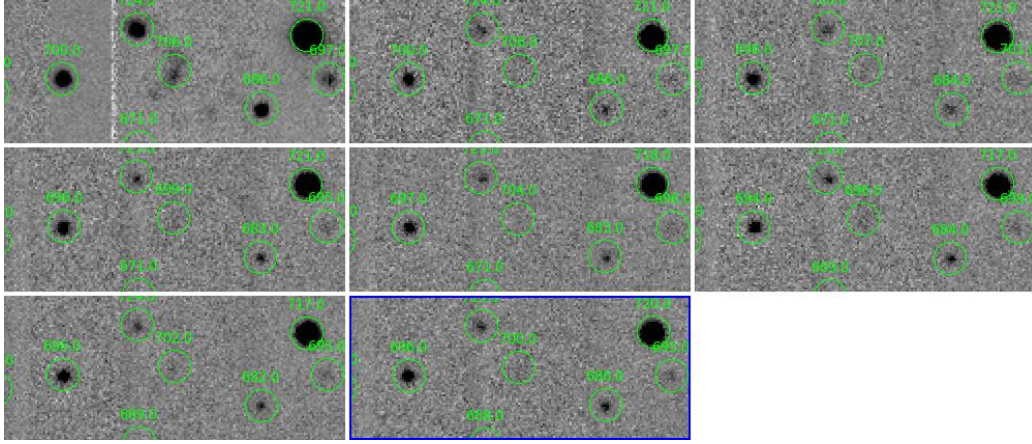
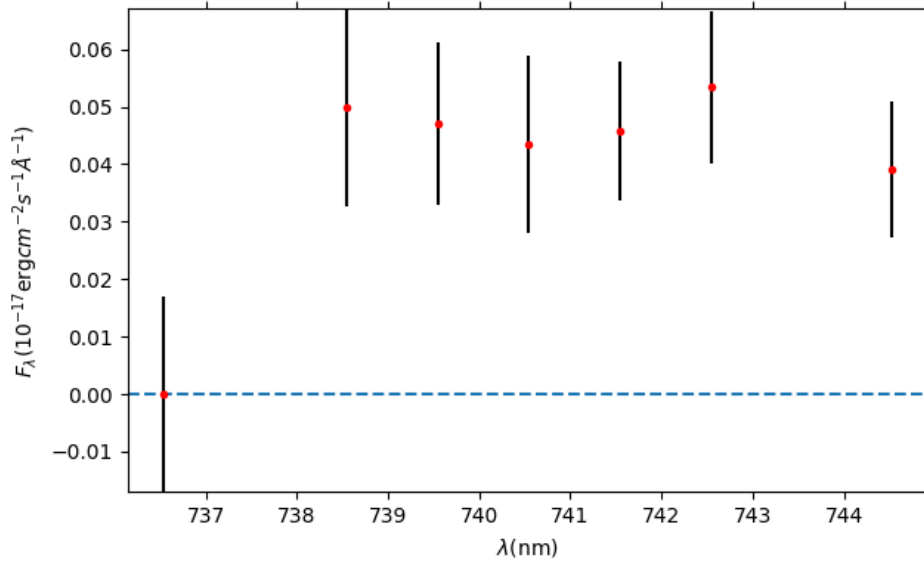
(a) *Finding chart. En el centro fuente número 303.*(b) *Distribución de flujos calibrados de la fuente número 303.*

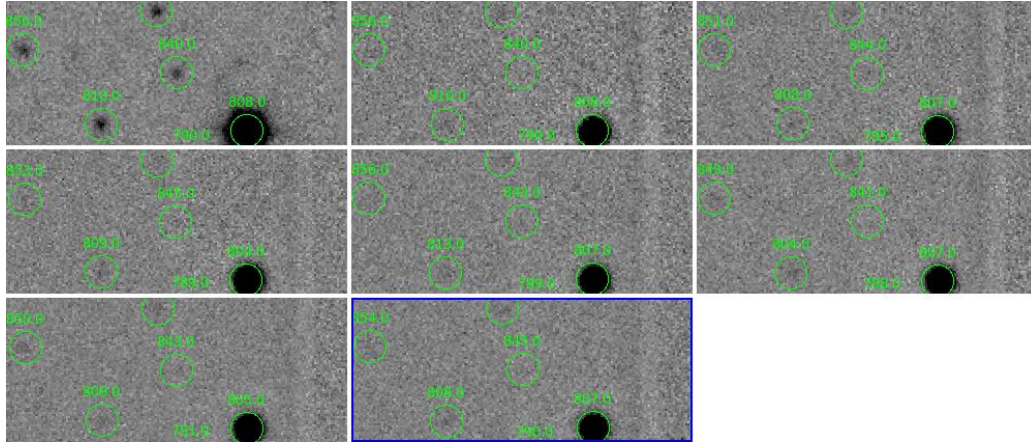
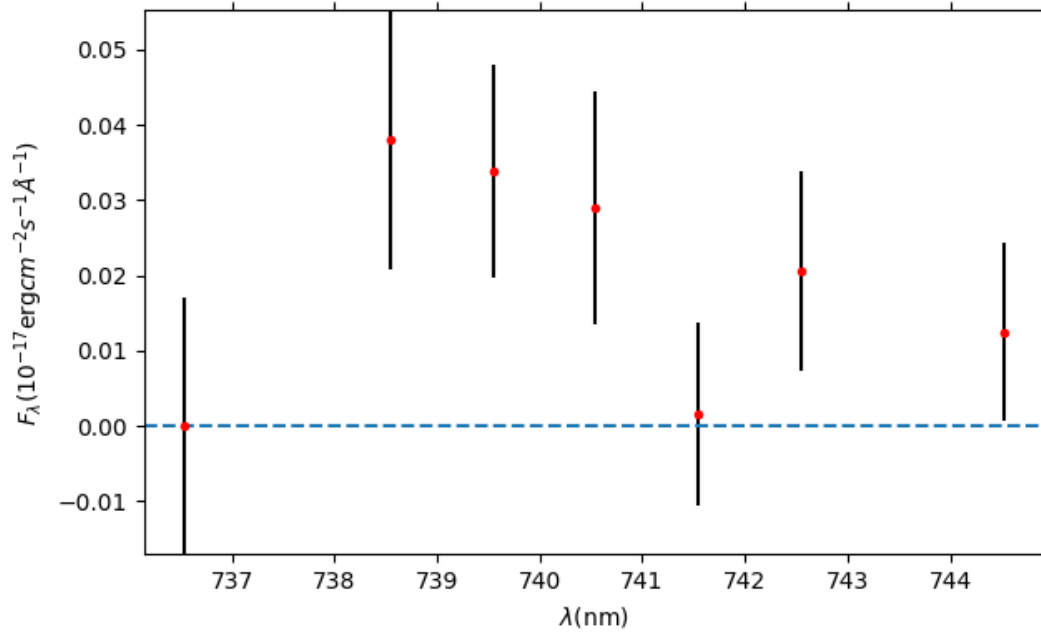
Figura 5.1: (a) *Finding chart* para la fuente número 303. Se muestran ocho imágenes correspondientes a cada uno de los ocho filtros. Se muestran ordenados de menor a mayor longitud de onda central de los filtros, desde el segundo de arriba hacia la derecha. La primera imagen arriba a la izquierda muestra la imagen obtenida con el filtro OS. (b) Espectro observado de esta fuente a través de los siete filtros. Cada punto rojo representa el flujo calibrado que ha atravesado un filtro. La línea discontinua azul indica el nivel del cero. (PR9).

5.3.2. Candidata aceptable: 401.

Otro caso a destacar es la fuente número 401, ver Fig. (5.2). Se trata de una fuente muy débil tal como puede observarse en su *finding chart*, Fig. (5.2.a), siendo realmente donde se aprecia con nitidez en la imagen conseguida con el filtro OS.

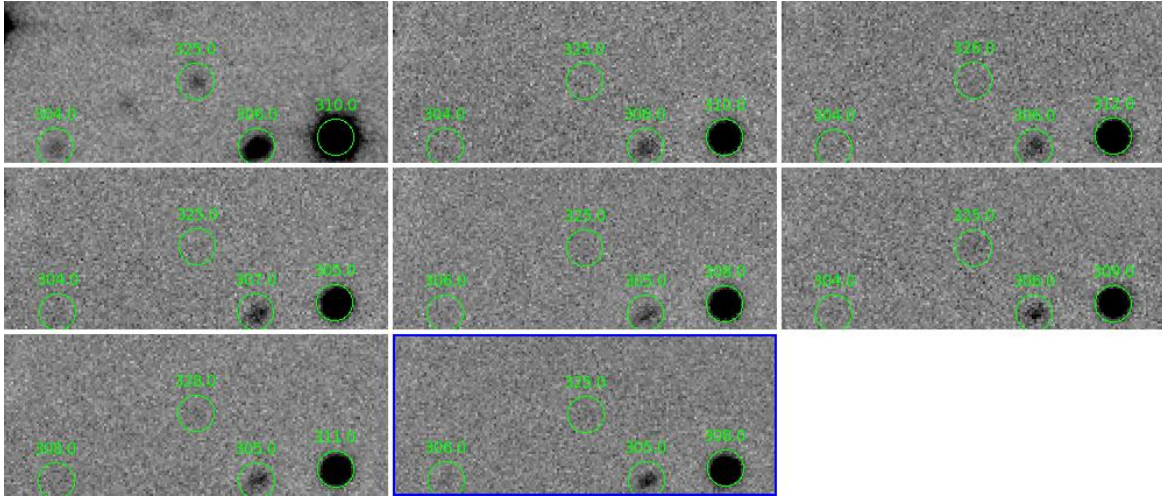
Ante la distribución de medidas en la Fig. (5.2.b), el método clasifica a esta fuente como una candidata a ser una galaxia de tipo 1. No obstante, a diferencia de lo que ocurre con la fuente 303, la zona plana y continua del filtro 2 en adelante, no está tan bien definida. Pero, hay que recordar que el método de selección creado es un método en primera aproximación que no contempla la detección por los filtros 5, 6 y 7. Por otro lado, el escalón entre el filtro 1 y el 2, aunque es pequeño es distinguible. Sin embargo, el escalón no es medible, precisamente por tratarse de un tipo 1, para los cuales el filtro 1 no detecta nada. En cuanto a las barras de error pese a que no dan tanta certeza como en la fuente 303, permiten discernir la forma propia de una fuente de tipo 1.

Esta fuente es un ejemplo de que si en una futura repetición de las observaciones estas tuvieran mejor un SNR, tiene muchas posibilidades de adquirir la seguridad de ser una candidata tan buena como la 303.

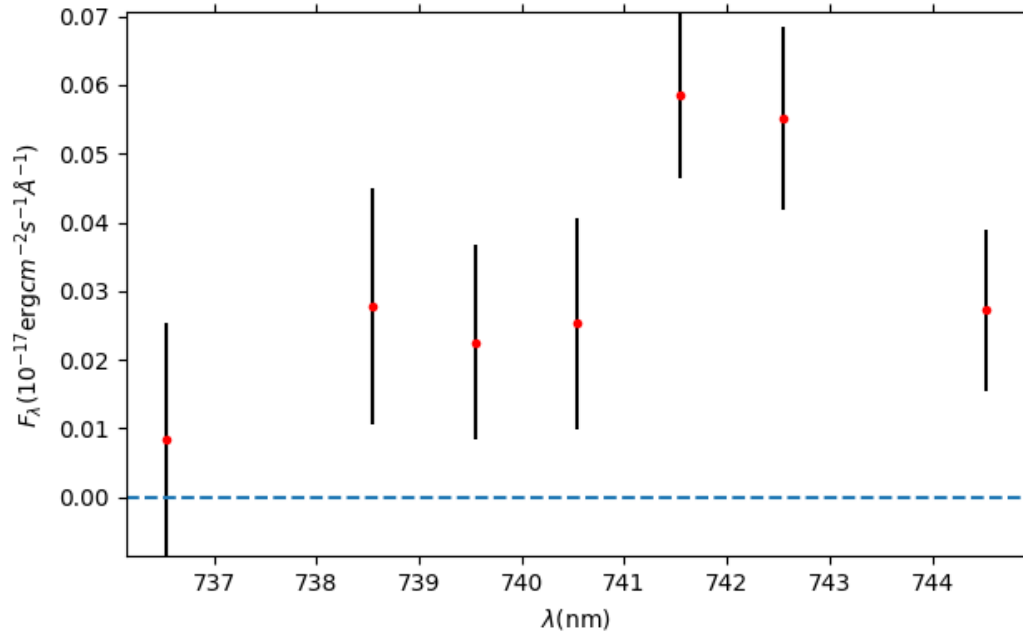
(a) *Finding chart. En el centro se observa fuente número 401.*(b) *Distribución de flujos calibrados de la fuente número 401.*Figura 5.2: *Finding chart para la fuente número 401. Disposición igual que en la Fig.(5.1). (PR9). .*

5.3.3. Cumplimiento del método de selección pero no de los criterios físicos: 58 y 623.

En cuanto a las fuentes 58 y 623 a pesar de encontrarse entre las mejores candidatas, estas no pueden considerarse como tal porque aunque cumplen las condiciones matemáticas del programa con el que se hace la selección, no se ajustan a lo esperado físicamente para este trabajo. Por otro lado, intuitivamente en ellas se ve, Fig. (5.3) y Fig. (5.4), la distribución de flujos que se busca. Por ello estas fuentes son a tener muy en cuenta ante una posible repetición de la toma de observaciones mejorando el cociente señal/ruido, así como ante un refinamiento del método de selección.

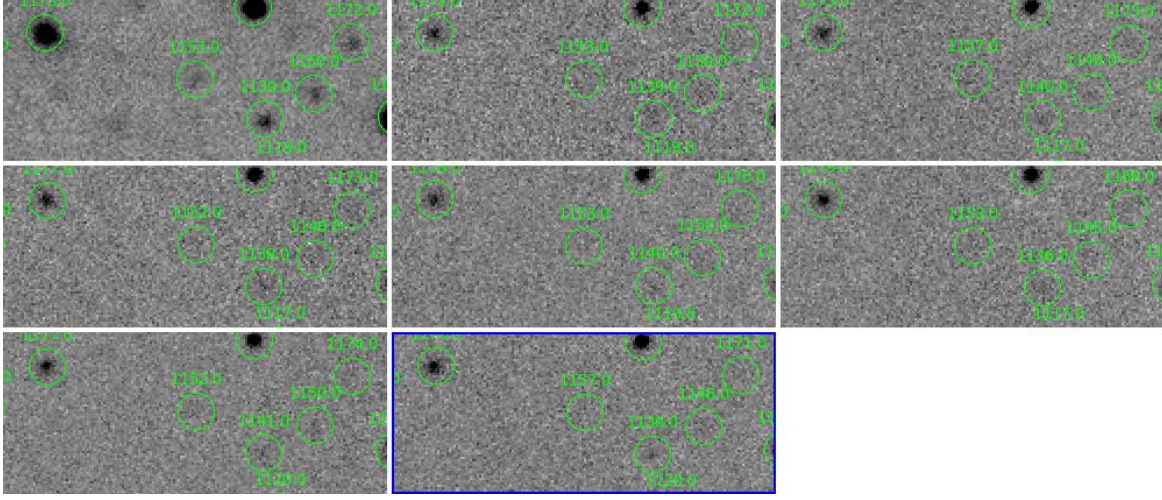
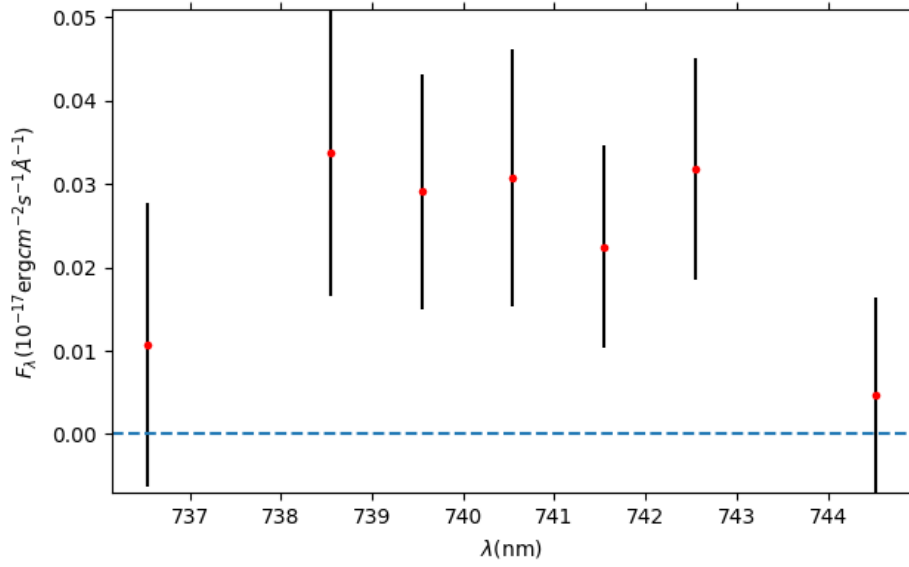


(a) Finding chart. En el centro fuente número 58.



(b) Distribución de flujos calibrados de la fuente número 58.

Figura 5.3: (a) Finding chart para la fuente número 58. El resto de descripción es la misma que en la Fig. (5.1). (PR9).

(a) *Finding chart. En el centro número 623.*(b) *Distribución de flujos calibrados de la fuente número 623.*Figura 5.4: (a) *Finding chart* para la fuente número 623. Disposición igual que en la Fig. (5.1). (PR9).

5.4. Limitaciones encontradas tras el análisis.

Incluyendo la condición más desfavorable inicial ya descrita, en el método de selección lo que se encuentra es que incluso en el caso más estricto, no podemos garantizar para todas las posibles candidatas que, del filtro 1 al filtro 2 haya un salto estadísticamente. Esto se traduce a que al observar los gráficos obtenidos con el método de selección de cada una de las 63 fuentes seleccionadas, las barras de error de los flujos calibrados son muy grandes. Con lo cual, parece que el SNR de los datos es insuficiente, y en base a la Ec. (3.6) se infiere que el tiempo de exposición es insuficiente.

En la propuesta (26) se había calculado que las galaxias tendrían un SNR de 3 o más.

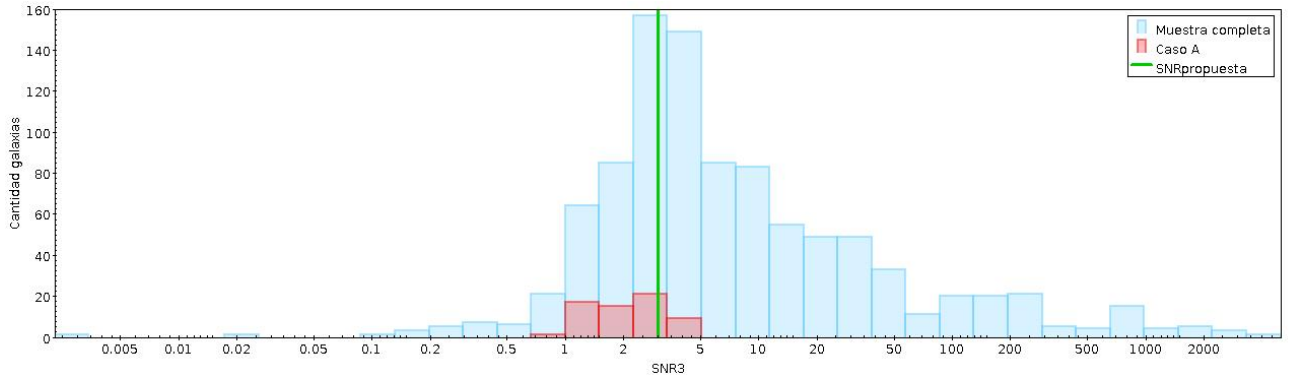


Figura 5.5: Distribución de la SNR de todas las fuentes a través del filtro 3, histograma en azul, junto con la muestra completa de fuentes seleccionadas del caso A Tab. (4.1), histograma en rojo. La línea verde marca el $SNR = 3$ aportado en la propuesta (26). (Topcat).

A la vista de los resultados de este trabajo, el SNR de la propuesta se ha sobreestimado y un reflejo de esto es lo aportado en la Fig. (5.5), donde se ve que la mayoría de las fuentes seleccionadas tienen un SNR menor al esperado, lo que apoya la conclusión anteriormente dicha. Con lo cual, aumentando el tiempo de exposición este cociente mejoraría y quizás permitiría discriminar un poco mejor las fuentes seleccionadas. Por otro lado, la Fig. (5.5), también permite ver que a pesar de que se haya pedido que los criterios físicos de este trabajo, favorezcan cantidades con barras de error grandes, la autora de este trabajo no se ha quedado con todas ellas si no que ha seleccionado una fracción pequeña de ellas luego realmente sí que se ha hecho una selección con criterio físico.

5.5. Tamaño físico del área observada frente al de un cúmulo.

Se ha calculado por medio de una calculadora cosmológica (29), la escala a $z = 5$ siendo $6.417 \text{ kpc}''$.

Por otra parte estimando el tamaño de un círculo que englobe aproximadamente la región buena creada (PR3), definida en la sección 3.1, se obtiene un radio de $\alpha_{reg} = 211.52''$.

Tomando estas dos cifras en conjunto la distancia física que corresponde con $211.52''$ a $z = 5$, es de 1.4 Mpc . Dado que el tamaño esperado de un cúmulo a $z = 5$ es de 4 Mpc (17), (6), la zona del cielo que estamos muestreando alrededor del cuásar es compatible con dicho tamaño.

5.6. Comparación con los resultados en submilimétricas.

Comparando los resultados obtenidos con el estudio realizado en submilimétricas (21), en los cuales se encontraron seis posibles candidatas, es de interés analizar si alguna de estas coincide con las encontradas en el presente trabajo. Para ello se superponen unas y otras.

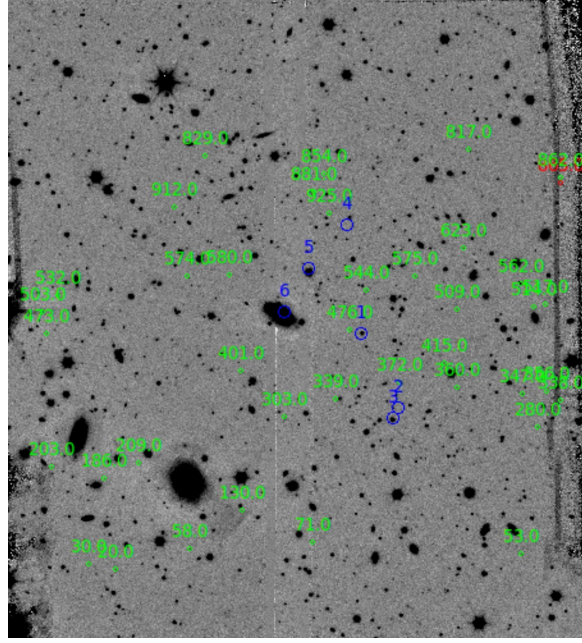


Figura 5.6: Imagen astronómica obtenida con el filtro OS. En ella se representan las fuentes seleccionadas con el método de selección del caso A, que engloba a las 63 fuentes seleccionadas, de la Tab. (4.1). Las galaxias de tipo 1 y 2 en verde, y las de tipo 3 y 4 en rojo. Al tiempo se representan las fuentes encontradas en submilimétricas, en azul, (21).

En la imagen de Fig. (5.6) se observa que, al representar las fuentes candidatas del estudio en submilimétricas junto con la muestra total de galaxias de 63 galaxias detectadas por el método presentado no se superpone ninguna. Pero esto no supone algo negativo para los resultados aportados en este trabajo porque, como ya se discutió en la introducción, para que una galaxia sea brillante en submilimétricas tiene que tener una alta formación estelar y una gran cantidad de polvo que oscurezca la luz de la galaxia. Por tanto, estas galaxias no tienen por qué ser más brillantes en el ultravioleta (en reposo), que es lo que se puede detectar con el método de este trabajo. Con lo cual, el hecho de que no halla superposición entre las galaxias encontradas por cada estudio, pone de manifiesto que este trabajo permite seleccionar galaxias que en submilimétricas no se están encontrando y que por tanto, hay más posibles candidatas de las que se encuentran en la búsqueda en submilimétricas (21).

Capítulo 6

Conclusiones y trabajo futuro.

A modo de síntesis se destaca como se han cumplido los objetivos del trabajo. Se ha hecho un trabajo de detección de fuentes minucioso en las imágenes astronómicas proporcionadas. Y se han detectado fuentes en cada filtro y en la imagen OS. Y se han identificado las fuentes comunes entre esas imágenes.

Posteriormente al trabajo de detección se ha realizado un calibrado fotométrico de las imágenes. En base a razonamientos físicos se ha desarrollado un método para seleccionar las fuentes al mismo z que el cuásar. Y se ha obtenido que por medio de este método se han encontrado varias candidatas. Además se ha comprobado que su distribución angular es compatible con el tamaño de un cúmulo al z del cuásar.

El bajo número de candidatas se puede atribuir, al menos en parte, a dos factores ajenos a este trabajo: por un lado el hecho de que los filtros no estén bien centrados, debido al error de SDSS al asignar el z de la fuente ($z = 5.1$ frente a $z = 5.04$ que se ha encontrado durante este trabajo). Por otro, las detecciones tienen un cociente señal/ruido bastante bajo, lo que no permite asegurar con alta certeza estadística que las fuentes estén efectivamente al z del cuásar.

Una repetición de las observaciones pero con una óptima sintonización de los filtros y con un mayor tiempo de exposición, permitiría comprobar si el método permite efectivamente seleccionar galaxias al mismo z que el cuásar de manera eficiente y estadísticamente significativa. Así por ejemplo, un siguiente paso podría ser el tomar espectros de las galaxias candidatas para comprobar que efectivamente son galaxias con un salto brusco. Y si se encontrasen suficientes, tanto hacia el rojo como al azul, como cada una tendría velocidades propias, se podría estimar la masa del cúmulo. Además, si entre ellas hay galaxias emisoras de Lyman- α , se podría estimar la tasa de formación estelar, que indicaría si estas galaxias están creciendo mediante formación estelar o no.

Bibliografía

- [1] ALFRED, P. The Sloan Digital Sky Survey, SDSS. (Copyright 2018). (Consulted on: 9 June 2019) <https://www.sdss.org/>.
- [2] BERTIN, E., AND ARNOUTS, S. Sextractor: Software for source extraction. *Institut d'Astrophysique Observatoire de Paris 117 v2.13* (June 1996), 393–404. doi:10.1051/aas:1996164 <https://www.astromatic.net/pubsvn/software/sextr>.
- [3] BRADLEY M, P. *An introduction to active galactic nuclei*. pp 200-204. University of Cambridge, (2003). ISBN:0-521-47348-9.
- [4] BRADT, H. *Astronomy Methods*, 1^a ed. The press syndicate of the university of Cambridge, Cambridge University, (2004). ISBN: 0-521-36440-X.
- [5] CABRERA-LAVERS, A. *OSIRIS-USER-MANUAL v3.1*. Grantecan S.A., (January 2014).
- [6] CHIANG, Y.-K., OVERZIER, R., AND GEBHARDT, K. Ancient Light from Young Cosmic Cities: Physical and Observational Signatures of Galaxy Proto-clusters. *The Astrophysical Journal* (December 2013). pp 779, 2. doi:10.1088/0004-637X/779/2/127.
- [7] EISBERG, R. M., AND RESNICK, R. *Quantum physics of atoms, molecules, solids, nuclei and particles; 2nd ed*. Wiley, New York, NY, 1985. <https://cds.cern.ch/record/105889>.
- [8] GALADI, D., AND GUTIERREZ, J. *Astronomía General. Teoría y práctica*. Parte VII in Omega, pp 787-792, 823-826, 837-839. EGEDSA, (2001). ISBN:84-282-1168-X.
- [9] IAC. Gran Telescopio de Canarias GTC. La Laguna (Santa Cruz de Tenerife), (2016). (Consulted on 9 May 2019) <http://www.gtc.iac.es/>.
- [10] IAC. Gran Telescopio de Canarias GTC. Instrumento OSIRIS. La Laguna (Santa Cruz de Tenerife), (2016). (Consulted on 9 May 2019) <http://www.gtc.iac.es/instruments/osiris/>.
- [11] IVISON, J. R., DUNLOP, J., SMAIL, I., DEY, A., LIU, C. M., AND GRAHAM, J. An excess of submillimeter sources near 4c 41.17: A candidate protocluster at $z = 3.8$? *The American Astronomical Society. All rights reserved. 542: 27E34* (December 2008), pp 27. Printed in U.S.A doi:10.1086/309536.
- [12] KASHIKAWA, N., KITAYAMA, T., DOI, M., MISAWA, T., KOMIYAMA, Y., AND OTA, K. The Habitat Segregation between Lyman Break Galaxies and Ly α Emitters around a QSO at $z \sim 5$. *The Astrophysical Journal* 663,2(10 July 2007), pp 765 – 773. doi : 10.1086/518410.

- [13] LANZ-OCA, L. F. *Detección de fuentes extragalácticas en mapas de la radiación del fondo cósmico de microondas mediante técnicas multifrecuenciales*. PhD thesis, Universidad de Cantabria, Santander, 2016.
- [14] MALCON S, L. *Galaxies Formation Astronomy Astrophysics Library*. Institute of Physics Publishing Bristol and Philadelphia, (2006). pp 38-41. ISBN: 978-3-540-73477-2.
- [15] MARK TAYLOR, E., AND ARNOUTS, S. Topcat: Tool for Operations on Catalogues And Tables. Version 4.6-3. *Astrophysics Group, Physics Department, University of Bristol* (9 May 2019). <http://www.star.bristol.ac.uk/~mbt/topcat/sun253/>.
- [16] MCMILLAN., S., AND , CHAISSON, E. *Astronomy Today*, 8^a ed. Pearson, 2003. pp 109-111 ISBN: 978-0-321-90167-5 (student edition).
- [17] MILLER, T. B., CHAPMAN, S. C., ARAVENA, M., ASHBY, M. L. N., HAYWARD, C. C., VIEIRA, J. D., WEISS, A., BABUL, A., BÉTHERMIN, M., AND BRADFORD, C. M. A massive core for a cluster of galaxies at a redshift of 4.3. *Nature* 556, 7702 (April 2018), 469–472. doi: 10.1038/s41586-018-0025-2.
- [18] MORTON, D. C. Atomic Data for Resonance Absorption Lines. I. Wavelengths Longward of the Lyman Limit. *The Astrophysical Journal Supplement Series, Vol: 77* (September 1991), 119–202. doi: 10.1086/191601.
- [19] NASA/ GSFC. The FITS Support Office. <https://fits.gsfc.nasa.gov/>.
- [20] PEEBLES, P. J. E. *Principles of physical cosmology*. Princeton in physics. Princeton University Press, United Kingdom, 1993. pp 39,500. ISBN: 0-691-07428-3.
- [21] PRIDDEY, R. S., IVISON, R. J., AND ISAAK, K. G. Environments of $z > 5$ quasars: searching for protoclusters at submillimetre wavelengths. *Monthly Notices of the Royal Astronomical Society* 383, 1 (January 2008), 289–296. doi: 10.1111/j.1365-2966.2007.12536.x.
- [22] SCISERVER. *SkyServer*. (2016) (Consulted on 25 April 2019) <http://skyserver.sdss.org/dr14/en/tools/chart/navi.aspx>.
- [23] SCISERVER. *Catálogo SDSS-DR14*. 2016. (Consulted on 25 April 2019) https://www.sdss.org/dr14/algorithms/qso_catalog/.
- [24] SHNEIDER, P. *Extragalactic Astronomy and Cosmology: An Introduction*. Springer. Springer Verlag Berlin Heidelberg 2006. ISBN-10 3-540-33154-3. (2006).
- [25] SIMBAD. Query by identifiers. Université de Strasbourg/CNRS. (Consulted on 26 June 2019) <http://simbad.u-strasbg.fr/simbad/sim-fid>.
- [26] STREBLYANSKA, A., AND CARRERA, F. *PETICION DE TIEMPO DE OBSERVACION (CAT) NOCTURNO (CAT NOCTURNO)*. Observatorios del Roque de los Muchachos y del Teide. La Laguna, Tenerife (Islas Canarias). IFCA, Septiembre 2010.
- [27] TIPPLER, A., AND PAUL MOSCA, G. *Física para la ciencia y la tecnología.*, 4 ed. Reverté, Vizcaya, (2004). Vol.II Cap 37. ISBN: 84-291-4384-X.

- [28] WRIGHT, E. L. Lyman alpha forest, Last modified: 1 August 2004. (Consulted on: 2 May 2019) <http://www.astro.ucla.edu/~wright/Lyman-alpha-forest.html>.
- [29] WRIGHT, E. L. Cosmological calculator. Last modified on 2005. <http://www.astro.ucla.edu/~wright/Lyman-alpha-forest.html>.

Apéndice A: Valores observados de los parámetros de selección.

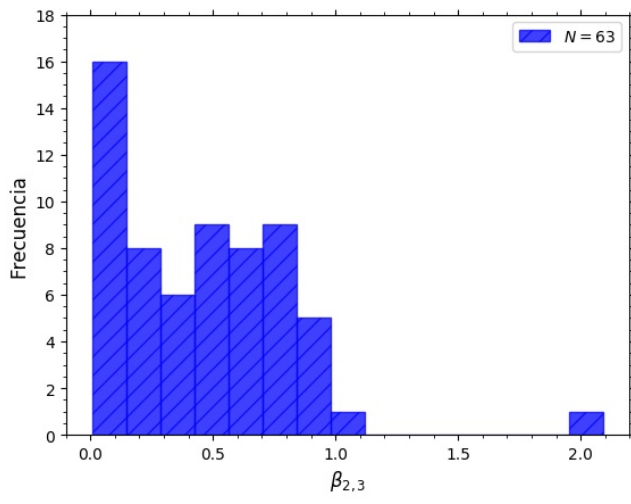
Se proporcionan a continuación los resultados obtenidos con las Ec. (4.1) Ec. (4.2) y Ec. (4.3) y con los valores límites establecidos para el caso A de la Tab. (4.1) para las 63 fuentes seleccionadas que contempla dicho caso,y, que engloban todas las fuentes del resto de casos.

Índice	$\alpha_{2,3}$	$\beta_{2,3}$	$\beta_{3,4}$	$\gamma_{1,2}$	$\gamma_{1,3}$
20	0.014	0.014	0.553	0.141	0.139
30	0.759	0.759	0.452	0.101	0.086
53	0.84	0.84	0.597	0.149	0.063
58	0.234	0.234	0.132	0.03	0.146
71	-0.57	0.57	0.54	0.128	0.043
88	0.751	0.751	0.861	1.194	2.059
104	-0.572	0.572	0.509	1.96	1.19
114	0.395	0.395	0.681	1.383	2.048
130	0.099	0.099	0.686	1.048	1.226
186	0.684	0.684	0.692	-	-
196	-0.155	0.155	0.519	1.632	1.482
203	-0.638	0.638	0.137	0.56	0.077
209	0.339	0.339	0.819	0.737	1.285
229	0.096	0.096	0.49	1.272	1.464
245	0.775	0.775	0.212	1.418	3.107
280	-0.06	0.06	0.501	0.457	0.408
292	0.756	0.756	0.222	1.952	3.377
301	0.138	0.138	0.285	1.194	1.424
302	-0.649	0.649	0.833	1.37	0.723
303	0.132	0.132	0.17	-	-
327	0.221	0.221	0.049	1.138	1.389
338	0.997	0.997	0.205	0.962	2.706
339	0.555	0.555	0.605	0.887	1.692
347	-0.006	0.006	0.931	0.145	0.148
356	-0.5	0.5	0.898	0.338	0.03
358	-0.68	0.68	0.149	1.677	0.845
360	-0.351	0.351	0.536	0.992	0.62

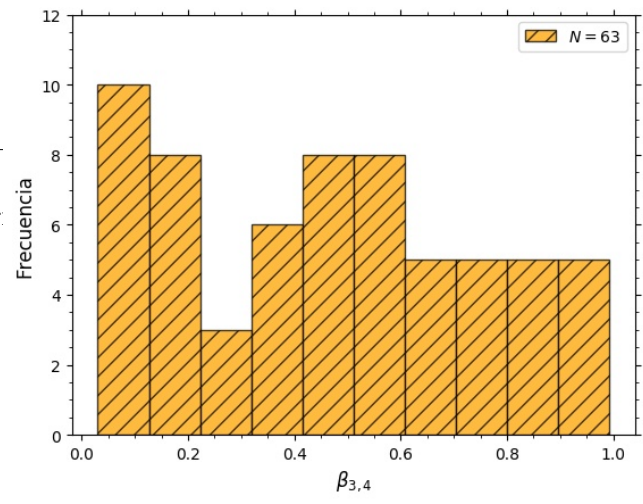
Tabla 6.1: Valores observados de los parámetros que se han usado para seleccionar las fuentes, junto con los casos de la Tab. (4.1) en los que se seleccionan, mostrando sólo los 63 del caso A (los otros casos son un subconjunto de éste). La primera columna, índice, indica el número de fuente.

Índice	$\alpha_{2,3}$	$\beta_{2,3}$	$\beta_{3,4}$	$\gamma_{1,2}$	$\gamma_{1,3}$
372	0.522	0.522	0.054	0.969	1.907
383	0.911	0.911	0.772	1.765	3.956
401	0.188	0.188	0.234	-	-
415	0.012	0.012	0.447	0.677	0.713
419	0.597	0.597	0.07	1.708	3.261
436	0.19	0.19	0.405	1.991	2.415
439	-0.009	0.009	0.885	1.549	1.637
473	0.92	0.92	0.38	0.191	0.058
476	0.043	0.043	0.32	0.152	0.183
498	0.963	0.963	0.461	1.134	2.855
503	0.529	0.529	0.118	-	-
505	0.04	0.04	0.551	1.524	1.748
509	0.465	0.465	0.775	0.417	0.801
514	0.007	0.007	0.746	0.9	0.96
517	-0.319	0.319	0.289	0.016	0.086
532	0.798	0.798	0.463	0.155	0.928
544	0.065	0.065	0.338	0.275	0.319
559	-0.079	0.079	0.03	1.898	1.885
562	-0.463	0.463	0.402	0.021	0.143
574	0.434	0.434	0.097	0.725	1.194
575	0.442	0.442	0.993	0.17	0.482
580	0.863	0.863	0.174	0.052	0.71
623	0.209	0.209	0.077	0.065	0.172
628	0.46	0.46	0.982	1.438	2.547
633	0.619	0.619	0.104	1.074	2.311
717	-0.248	0.248	0.801	1.924	1.67
720	-0.412	0.412	0.714	1.703	1.235
817	0.029	0.029	0.035	0.632	0.672
823	-0.9	0.9	0.392	1.664	0.661
829	0.039	0.039	0.702	1.006	1.099
854	-0.788	0.788	0.934	0.474	0.03
862	-0.837	0.837	0.114	-	-
865	2.094	2.094	0.769	0.329	0.16
881	-0.249	0.249	0.467	-	-
912	0.816	0.816	0.535	0.038	0.451
925	0.303	0.303	0.66	0.037	0.083

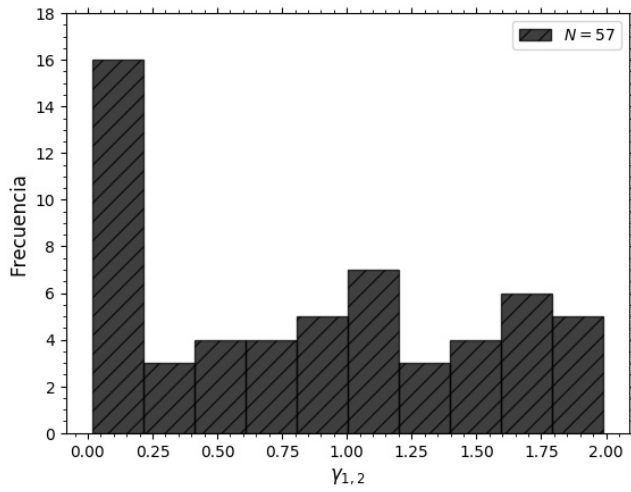
Tabla 6.2: *Continuación de Tab.(6.1)*



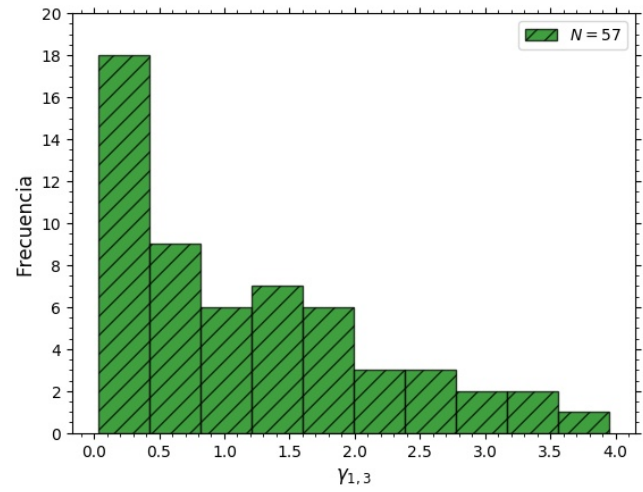
(a)



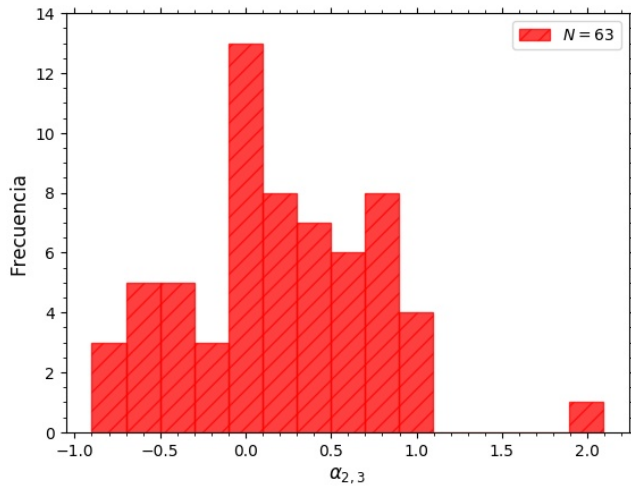
(b)



(c)



(d)



(e)

Figura 6.1: Histogramas asociados con los valores en las tablas (6.1) y (6.2).

Apéndice B: Descripción breve de programas y scripts. Y códigos.

Los programas que se han creado durante este trabajo se han realizado con el lenguaje de programación Python versión 3.6, y, en código bash. Todos ellos pueden ser corridos desde una terminal de Ubuntu. En concreto este trabajo se ha realizado en Ubuntu 18.04.

A continuación se da una breve explicación de los principales códigos que se han desarrollado durante este trabajo. Y, posteriormente, se proporcionan los códigos.

Como apunte técnico, cabe decir que con la elección de Python como lenguaje de programación se ha buscado la facilidad en el código en un contexto donde la eficiencia no resulta uno de los puntos más importantes. En efecto, puede decirse que las instrucciones de Python así como de código bash, se ejecutan de forma instantánea y es la generación de *finding charts* las que suponen más tiempo, afectando al tiempo total de ejecución de los programas.

PR	Nombre	Descripción
PR0	imagenSimulada.py	Lee una imagen astronómica y la lista de fuentes asociada a ella. Y escribe una imagen con la astrometría de la imagen simulada de entrada pero con las fuentes simuladas en las posiciones de que la lista alberga.
PR1	compruebaImagenesSimuladas.py	Genera un fichero .reg para ser usado en ds9, con las fuentes en cada imagen simulada. Y dicho fichero se superpone con el de las imágenes reales comprobando así que hay superposición de las fuentes.
PRF1	allMainFase1.sh	Script que contiene programas y llamadas a otros programas para el desarrollo de la fase 1.
PR2	regionComunCatalogos.py	Crea una región buena, libre de defectos, común a todas las imágenes astronómicas. Tal región engloba las fuentes con las que se hace el estudio en el resto del trabajo.
PR3	marcaCircFuentes.py	Crea un fichero con regiones circulares dentro de la región buena y común para indicar las fuentes detectadas por SExtractor en dicha región.
PR4	tmatchn	Comando de topcat que asocia las fuentes medidas en cada imagen a las posiciones de las fuentes detectadas en la imagen OS.
PR5	findingCharts.py	Genera un archivo .sh con las instrucciones para generar las <i>findingCharts</i> en DS9. Dicho fichero es el siguiente.
PR5.1	generaFindingCharts.sh	Archivo con instrucciones para ser entendidas por el DS9.
PRF2	allMainFase2.sh	Script con los programas necesarios en la fase 2. Su ejecución desarrolla toda la fase 2.
PR6	calibracionConEstrella.py	Calibra los flujos de las fuentes detectadas. Genera gráficos y archivos de resultados calculados.
PR7	suavizadoEspectro.py	Suaviza el espectro del flujo observado del cuásar con un filtro. Y grafica el espectro suavizado sobre el del flujo observado sin suavizar.

Tabla 6.3: Descripción breve de cada uno de los programas o scripts para la realización de este trabajo. En la primera columna se da la etiqueta con la cual se hace referencia a cada uno a lo largo del presente documento.

PR	Nombre	Descripción
PR8	medirZlineaSiIV.py	Hace un modelo de ajuste al espectro del cuásar a partir de las línea SiIV y CIV. Proporciona la longitud de onda de la línea del SiIV y a partir de ella calcula el z del cuásar.
PRF3	allMainFase3.sh	Script que recopila todos los programas necesarios en la fase 3. Su ejecución desarrolla la fase 3.
PR9	metodoSeleccion.py	Selecciona y clasifica las fuentes detectadas en base a criterios físicos establecidos. Genera gráficos de los espectros observados para todas las fuentes.
PR10	asociaFindingChartFuente.py	Asocia la <i>finding chart</i> de cada fuente con su gráfica correspondiente.
PR11	diagramasCriteriosFisicos.py	Genera gráficos para entender de forma ilustrativa los criterios físicos aplicados en el método de selección.
PR12	histogramas.py	Crea histogramas para ilustrar los valores observados a partir de los parámetros con los que se hace la selección de las 63 galaxias del caso A (sección 4 Tab. (4.1)).
PR13	superponeEstudios.py	Abre la región con las fuentes encontradas en el presente trabajo y la de las encontradas en submilimétricas en otro estudio (21). Superpone ambas regiones.
PR14	fuentesAsuperponer.py	Crea regiones para las fuentes seleccionadas por el método de selección. Distingue con colores si son LAB o LAE.

Tabla 6.4: Continuación de la Tab. (6.3).

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on May 2019
5 @author: Maria Ruiz Perez
6 """
7 #####
8 ##                                PRO                                imagenSimulada.py                                ##
9 ##-----##
10 ## Lee una imagen y una lista de fuentes y escribir una imagen ##
11 ## con la astrometria de la imagen de entrada y las fuentes falsas ##
12 ## en las posiciones de la lista. ##
13 #####
14 import sys
15 import numpy as np
16 from astropy.io import fits
17
18
19 print('\n\n Comienza imagenSimulada.py\n\n')
20
21 #-----
22 ### Lectura de parametros de entrada.
23 #-----
24 if len(sys.argv) != 4:
25     print (" ")
26     print (" ERROR: need 3 command line arguments: ")
27     print ("          una imagen FITS de referencia, de entrada.")
28     print ("          una lista de posicines X,Y de entrada.")
29     print ("          un nombre de archivo de la imagen FITS de salida con la
          astrometria de las fuentes primeras y falsas en las posiciones X, Y")
30     exit(1)
31
32 # sys.argv[0] es el nombre del archivo python que se est  ejecutando
33 infile1=sys.argv[1]
34 infile2=sys.argv[2]
35 outfile=sys.argv[3]
36 #
37
38 print("\n")
39 print(" Nombre del archivo de referencia de entrada: "+infile1)
40 print(" Lista de posiciones X,Y de entrada: "+infile2)
41 print(" Nombre de fihero de salida: "+outfile)
42 print("\n")
43
44
45 #-----
46 ### Lectura de la lista de fuentes y guardado de posiciones
47 ### en el entero mas cercano.
48 #-----
49 soulist=np.loadtxt(infile2,comments="#")
50 x=np rint(soulist[:,0]).astype('int')
51 y=np rint(soulist[:,1]).astype('int')
52
53
54 #-----
55 ### Leyendo imagen en

```

```
56 #-----
57 print(" ")
58 file1=fits.open(infile1)
59 #data1=np.asarray(file1[0].data,dtype=np.float64)
60 data1=file1[0].data
61 print("      Size of image ",data1.shape)
62 nx=data1.shape[0]
63 ny=data1.shape[1]
64
65 #-----
66 ### Ahora imagen falsa.
67 #-----
68 # A adiando fuentes
69 data=np.zeros(data1.shape)
70 for i in range(len(x)):
71     j=x[i]-1
72     k=y[i]-1
73     if (j<nx and k<ny):
74         data[k,j]=50.0
75 # Fin del bucle
76 # Filtro gaussiano
77 from scipy.ndimage import gaussian_filter
78 data=gaussian_filter(data,sigma=1.0,mode='constant')
79
80 # gaussian noise, ave,rms from sextractor run on os_median
81 ave=0.000450942
82 rms=0.010167
83 data +=np.random.normal(loc=ave,scale=rms,size=data.shape)
84
85
86
87 #-----
88 ### Escribe imagen de salida.
89 #-----
90 hdu=fits.PrimaryHDU(data)
91 hdu.header=file1[0].header
92 hdu.writeto(outfile,overwrite=True)
93
94
95
96 print('\n\n Finished fakeimage.py\n\n')
97
98 exit(0)
99
100 ## Fin imagenSimulada.py -----
```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on May 2019
5 @author: Maria Ruiz Perez
6 """
7
8 #####
9 ##          PR1          compruebaImágenesSimuladas.py          ##
10 ##-----##
11 ## Comprueba que las imágenes simuladas creadas con          ##
12 ## imagenSimulada.py son correctas. Para serlo, las fuentes de ##
13 ## cada imagen original FITS han de encontrarse en la misma    ##
14 ## posición que en la imagen simulada asociada.                ##
15 #####
16
17 #-----
18 ###  Obtiene el catálogo a partir de la imagen OS
19 #-----
20
21 sextractor m0s_8.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -PHOT_APERTURES 5 -
   CATALOG_NAME os_median.cat
22
23
24 #-----
25 ### Genera un fichero de regiones circulares empleando el catálogo de la imagen os.
26 ### Del fichero de entrada tomo las columnas 4, 5 y 1 en formato float(%f),
27 ### float(%f) y string(%s).
28 ##          arch. entrada: os_median.cat
29 ##          arch. salida: os_median.reg
30 #-----
31
32 grep -v "#" os_median.cat | awk -F' ' '{printf("fk5;circle(%f,%f,2\\") # label={%s}\\n
   ",$4,$5,$1)}' > os_median.reg
33
34
35 #-----
36 ###  HACER MANUALMENTE:
37 #- Ejecutar cada una de las siguientes líneas de código para abrir el ds9 para
38 #  cada imagen 736,738,... con las regiones os anteriores.
39 #- Guardar las regiones en coordenadas X Y (físicas) con el nombre que se indica
40 #  después de cada línea de código siguiente:
41 #- Cuidado con crear más regiones de las que son.
42 #-----
43
44 ds9 m736_1.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25
45 #  nombre: os_median_736xy.reg
46 ds9 m738_2.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25
47 #  nombre: os_median_738xy.reg
48 ds9 m739_3.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25
49 #  nombre: os_median_739xy.reg
50 ds9 m740_4.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25

```

```

51 # nombre: os_median_740xy.reg
52 ds9 m741_5.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25
53 # nombre: os_median_741xy.reg
54 ds9 m742_6.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25
55 # nombre: os_median_742xy.reg
56 ds9 m744_7.fits -geometry 1000x1000 -invert -region load os_median.reg -scale linear
   -scale mode zscale -scale scope local -zoom to 0.25
57 # nombre: os_median_744xy.reg
58
59
60
61 #-----
62 ### Genera un fichero donde guarda la coord x e y de cada
63 ### imagen 736, 738, etc...
64 # arch. entrada: ficheros creados manualmente antes.
65 # arch. salida: ficheros os_median_***.lis
66 #-----
67
68 grep "label" os_median_736xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_736.lis
69 grep "label" os_median_738xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_738.lis
70 grep "label" os_median_739xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_739.lis
71 grep "label" os_median_740xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_740.lis
72 grep "label" os_median_741xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_741.lis
73 grep "label" os_median_742xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_742.lis
74 grep "label" os_median_744xy.reg | sed -e"s/circle(/ /1" | awk -F',' '{print $1" "$2}'
   > os_median_744.lis
75 #
76
77
78 #-----
79 ### Genera una imagen ficticia *.fits para cada imagen real
80 ### 736,738,739... empleando el programa python imagenSimulada.py
81 #-----
82
83 python fakeimage.py m736_1.fits os_median_736.lis os_median_736_fake.fits
84 python fakeimage.py m738_2.fits os_median_738.lis os_median_738_fake.fits
85 python fakeimage.py m739_3.fits os_median_739.lis os_median_739_fake.fits
86 python fakeimage.py m740_4.fits os_median_740.lis os_median_740_fake.fits
87 python fakeimage.py m741_5.fits os_median_741.lis os_median_741_fake.fits
88 python fakeimage.py m742_6.fits os_median_742.lis os_median_742_fake.fits
89 python fakeimage.py m744_7.fits os_median_744.lis os_median_744_fake.fits
90 #
91
92
93 #-----
94 ### Comprobacion.
95 ### Genera catalogos con la imagen ficticia.
96 #-----

```



```

97
98 sextractor os_median_736_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_736_fake.cat
99 sextractor os_median_738_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_738_fake.cat
100 sextractor os_median_739_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_739_fake.cat
101 sextractor os_median_740_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_740_fake.cat
102 sextractor os_median_741_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_741_fake.cat
103 sextractor os_median_742_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_742_fake.cat
104 sextractor os_median_744_fake.fits -DETECT_MINAREA 10 -DETECT_THRESH 2 -
    PHOT_APERTURES 5 -CATALOG_NAME os_median_744_fake.cat
105 #
106
107 #-----
108 ### Genera regiones x de los catalogos de la imagen ficticia.
109 #-----
110
111 grep -v "#" os_median_736_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_736_fake.reg
112 grep -v "#" os_median_738_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_738_fake.reg
113 grep -v "#" os_median_739_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_739_fake.reg
114 grep -v "#" os_median_740_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_740_fake.reg
115 grep -v "#" os_median_741_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_741_fake.reg
116 grep -v "#" os_median_742_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_742_fake.reg
117 grep -v "#" os_median_744_fake.cat | awk -F' ' '{printf("fk5;point(%f,%f) # point=x
    color=red label={%s}\n",$4,$5,$1)}' > os_median_744_fake.reg
118 #
119
120
121 #-----
122 ### Abre el ds9 para las imagenes reales, empleando las regiones de la imagen
123 ### ficticia.Si la x marca las fuentes de las imagenes 736,738...
124 ### el procedimiento realizado es correcto.
125 #-----
126
127 ds9 m736_1.fits -geometry 1000x1000 -invert -region load os_median_736_fake.reg -
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -
    zoom to 0.25
128
129 ds9 m738_2.fits -geometry 1000x1000 -invert -region load os_median_738_fake.reg -
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -
    zoom to 0.25
130
131 ds9 m739_3.fits -geometry 1000x1000 -invert -region load os_median_739_fake.reg -
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -
    zoom to 0.25
132

```

```
133 ds9 m740_4.fits -geometry 1000x1000 -invert -region load os_median_740_fake.reg -  
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -  
    zoom to 0.25  
134  
135 ds9 m741_5.fits -geometry 1000x1000 -invert -region load os_median_741_fake.reg -  
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -  
    zoom to 0.25  
136  
137 ds9 m742_6.fits -geometry 1000x1000 -invert -region load os_median_742_fake.reg -  
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -  
    zoom to 0.25  
138  
139 ds9 m744_7.fits -geometry 1000x1000 -invert -region load os_median_744_fake.reg -  
    region load os_median.reg -scale linear -scale mode zscale -scale scope local -  
    zoom to 0.25  
140 #  
141  
142 ## Fin comprobacion -----
```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on May 2019
5 @author: Maria Ruiz Perez
6 """
7 #####
8 ## PRF1      MAIN 1- FASE 1:      DETECCION DE FUENTES Y EXTRACCION      ##
9 ##                                     DE DATOS DE LAS IMAGENES ASTRONOMICAS .FITS      ##
10 ##-----##
11 ## Archivo con el que se realiza toda la fase 1. Registro de los scripts creados ##
12 ## por el orden en que deben ser ejecutados en la terminal. Hace llamadas a otros ##
13 ## programas y contiene lineas de codigo propias.      ##
14 #####
15 """
16 Ejecuta PR1
17 Corre Sextractor con las imagenes simuladas. Sextractor detecta las fuentes en la
    primera imagen, imagen simulada, (tomada de referencia) de cada linea de codigo y
    en la segunda, asociada a un filtro, mide los flujos en las posiciones de las
    fuentes detectadas.
18
19 Retorna: catalogos de fuentes.
20 (Informacion de cada catalogo: numero de fuentes, coordenas X y Y en pixeles,
    ascension recta y declinacion en grados,
21 flujo y error, y cuentas.)
22 """
23 # Nombres de imagenes de referencia *fake.fits
24 # Nombre imagenes donde mide los flujos: m***.fits
25 # Nombre del archivo donde se guardan las fuentes seleccionadas con sextractor: m***.
    cat
26 # Parametros fijados para Sextractor: -DETECT_THRESH = 2 -DEBLEND_MINCONT = 0.001 -
    PHOT_APERTURES = 4
27
28 sextractor simulada_median736.fits,m736_1.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m736.cat
29 sextractor simulada_median738.fits,m738_2.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m738.cat
30 sextractor simulada_median739.fits,m739_3.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m739.cat
31 sextractor simulada_median740.fits,m740_4.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m740.cat
32 sextractor simulada_median741.fits,m741_5.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m741.cat
33 sextractor simulada_median742.fits,m742_6.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m742.cat
34 sextractor simulada_median744.fits,m744_7.fits -c default.sex -DETECT_THRESH 2 -
    DEBLEND_MINCONT 0.001 -PHOT_APERTURES 4 -CATALOG_NAME m744.cat
35
36 ##-----
37
38 """
39 Invoca a PR2.
40 Nombre: regionComunCatalogos.py
41
42 Requiere: regiones .reg para cada imagen m***.fits
43 Retorna: un catalogo, en dos formatos, de las fuentes de cada catalogo que se

```

```

encuentren en una region comun, a todas las regiones, libre de impurezas.
44 Nombre archivo salida 1 : regCuadMin.cat (contiene los datos en columnas.)
45 Nombre archivo salida 2 : regCuadMin.reg (en formato ds9, lo puedes abrir en ds9
)
46
47 """
48
49 python regionComunCatalogos.py regionComun.reg regionComun.cat
50
51
52 """
53 Invoca PR3.
54 Nombre: marcaCircFuentes.py
55
56 Requiere:
57     argumento [0] = nombre del archivo python
58     argumento [1] = archivo sextr con todas las fuentes encontradas en la imagen
    fits correspondiente.
59     argumento [2] = datos en columnas de la min. comun region rectangular
60 Retorna:
61     argumento [3] = region circular de cada imagen
62     argumento [4] = coordenadas en grados de la dec. asen. rec de fuentes dentro
    de la minima comun region
63     argumento [5] = define la cabecera de cada archivo de salida del argumento
    [4]
64
65 """
66 #
67 python marcaCircFuentes.py m736.cat regionComun.cat m736Circ_1.reg m736_1.topcat 1
68 #
69 python marcaCircFuentes.py m738.cat regionComun.cat m738Circ_2.reg m738_2.topcat 2
70 #
71 python marcaCircFuentes.py m739.cat regionComun.cat m739Circ_3.reg m739_3.topcat 3
72 #
73 python marcaCircFuentes.py m740.cat regionComun.cat m740Circ_4.reg m740_4.topcat 4
74 #
75 python marcaCircFuentes.py m741.cat regionComun.cat m741Circ_5.reg m741_5.topcat 5
76 #
77 python marcaCircFuentes.py m742.cat regionComun.cat m742Circ_6.reg m742_6.topcat 6
78 #
79 python marcaCircFuentes.py m744.cat regionComun.cat m744Circ_7.reg m744_7.topcat 7
80
81 ##-----
82
83 """
84 Ejecuta PR4
85 Combina todos los ficheros .topcat asociados a cada cat logo.
86 Requiere: archivos .topcat generados por PR
87 Retorna: nico archivo .topcat: newOS_combinacionTopcatV2_match.topcat
88
89 """
90
91 topcat -stilts tmatchn multimode=group nin=7 matcher=sky params="1" fixcols=all in1=
    m736_1.topcat ifmt1=ascii values1="A1 D1" in2=m738_2.topcat ifmt2=ascii values2="
    A2 D2" in3=m739_3.topcat ifmt3=ascii values3="A3 D3" in4=m740_4.topcat ifmt4=ascii
    values4="A4 D4" in5=m741_5.topcat ifmt5=ascii values5="A5 D5" in6=m742_6.topcat

```

```

ifmt6=ascii values6="A6 D6" in7=m744_7.topcat ifmt7=ascii values7="A7 D7" join1=
match join2=match join3=match join4=match join5=match join6=match join7=match ocmd
='addcol index "index"' out=combinaFuentesCatalogos.topcat ofmt=ascii
92
93 ##-----
94
95 """
96 Invoca PR5.
97 Nombre:   findingCharts.py
98 Requiere: combinaFuentesCatalogos.topcat
99 Retorna:  generaFindingCharts.sh, archivo con las instrucciones para ser entendidas
           en ds9 y generar las finding charts
100
101 """
102 python findingCharts.py combinaFuentesCatalogos.topcat generaFindingCharts.sh
103
104 # Lee el fichero resultante anterior para obtener 8 finding chart por imagen generada
   en ds9.
105
106 sh generaFindingCharts.sh
107
108 ##-----
109
110 # Comando para aadir una columna con nombre de las fotos al archivo topcat
   resultante de la combinaci n.
111 topcat -stilts tpipe in=newOS_combinacionTopcatV2_match.topcat ifmt=ascii cmd='addcol
   rutaFotos "\"/home/maria/Escritorio/TFGcodigos/consola/foto\":"+padWithZeros(index
   ,4)+\".jpg\""' out=combinaFuentCatalogYrutaFindingCharts.topcat ofmt=ascii
112
113
114 ## FIN allMainFase1.sh -----

```

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on March 2019
5
6 @author: Maria Ruiz Perez
7 """
8
9 #####
10 ##                                PR2                                Nombre: regionComunCatalogos.py                                ##
11 ##-----##
12 ## Programa que determina la minima comun region de un conjunto de regiones ##
13 ## rectangulares. ##
14 ## Retorna: ##
15 ##     -Archivo que contiene los parametros de la minima region rectangular ##
16 ##     en columnas. ##
17 ##     -Archivo escrito en formato para ser entendido por ds9 tambien de la ##
18 ##     minima region rectangular. ##
19 #####
20
21
22 import math
23
24
25 # Nombres de los archivos a leer:
26 nombres = ["regCuad736_1", "regCuad738_2", "regCuad739_3", "regCuad740_4", "
           regCuad741_5", "regCuad742_6", "regCuad744_7"]
27
28 # Guarda las lineas box(...)
29 linesBox = []
30
31 # Ascencion recta maxima.
32 ARMax = []
33 # Ascencion recta minima
34 ARMin = []
35 # Declinacion recta maxima.
36 DecMax = []
37 # Declinacion recta minima.
38 DecMin = []
39
40 # Coordenadas de todas las cajas.
41 coord = []
42
43 # Lee todos los archivos:
44 for i in range(len(nombres)):
45     # Accede a cada archivo
46     file = open(nombres[i]+".reg", 'r')
47     # Guarda todas las lineas de texto del fichero.
48     contenido = file.readlines()
49     #(contenido es una lista, cada elemento es una linea del fichero.)
50
51     # Busca la linea donde esta box(..) y la guarda:
52     box = contenido[3]
53
54     # Selecciona del conjunto de caracteres la parte numerica.
55     box = box.split('box')
```

```

56     box = box[1].split(',')
57     box = box[1].split(')')
58     box = box[0].replace('"', ' ')
59     box = box.split(',')
60
61     # Convierte el string numerico en numeros float.
62     box = list(map(float,box))
63
64     # Convierte los segundos a grados.
65     box[2] = box[2]/3600
66     box[3] = box[3]/3600
67
68     # A ade los datos en bruto de cada region a una lista.
69     linesBox.append(box)
70
71     # Coloca en las listas siguientes los coordenadas de los vertices
72     # de las regiones.
73     # Calcula la declinacion minima y maxima de cada region.
74     decMax = (box[1]+box[3])/2)
75     decMin = (box[1]-box[3])/2)
76
77     # Calcula la correccion en la ascension recta.
78     correccion = math.cos(math.radians(box[1]-box[3]/2))
79     box[2] = box[2]/correccion
80
81     # Calcula la ascension recta maxima y minima de cada region.
82     arMax = (box[0]+(box[2])/2)
83     arMin = (box[0]-(box[2])/2)
84
85     ## Guarda en listas los parametros de todas las cajas rectangulares de
86     ## cada imagen fits. (Estas cajas son las verdes en ds9.)
87     DecMax.append(decMax)
88     DecMin.append(decMin)
89     ARMax.append(arMax)
90     ARMin.append(arMin)
91
92     #-----
93     ### 1  archivo  a generar:
94     #-----
95     ## Genera un archivo donde guardara los datos de la minima region:
96     outFile = open("regCuadMinV3.cat", "w")
97
98     # Escribe la cabecera del archivo.
99     outFile.write("#ARMax      ARMin      DecMax      DecMin \n")
100    outFile.write(str(min(ARMax))+ " "+str(max(ARMin))+ " "+str(min(DecMax))+ " "+str(max(
101        DecMin)))
102
103    #-----
104    ### 2  archivo  a generar:
105    #-----
106    ## Arhivo con la region cuadrada en formato ds9:
107    DecPosMed = (min(DecMax)+max(DecMin))/2
108    Declong = (min(DecMax)-max(DecMin))*3600
109    ArPosMed = (min(ARMax)+max(ARMin))/2
110    Arlong = (min(ARMax)-max(ARMin))*3600
111

```

```

111 ## Tiene que estar en el formato que se pone a continuacion para que pueda ser leido
    por ds9.
112 # El color elegido de la region cuadrada minima es rojo.
113 outFileDs9 = open("regCuadMinV3.reg", "w")
114 outFileDs9.write("fk5;box("+str(ArPosMed)+","+str(DecPosMed)+","+str(Arlong)+','+str
    (Declong)+','+str(0) # color=red \n')
115
116
117 ## Fin regionComunCatalogos.py -----

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on March 2019
5
6 @author: Maria Ruiz Perez
7 """
8
9 #####
10 ## PR3 Nombre: marcaCircFuentes.py ##
11 ##-----##
12 ## Programa para leer un fichero de salida de sextractor y escribir dos ##
13 ## ficheros: ##
14 ## Un fichero region, .reg, con las regiones circulares y un fichero ##
15 ## topcat, .topcat, de las fuentes dentro de la minima comun region. ##
16 ## Ademas, las fuentes elegidas se encuentran dentro de la region comun ##
17 ## cuadrada que ya se debe haber definido previo a la ejecucion de ##
18 ## este programa. ##
19 ## ##
20 ## Requiere: ##
21 ## -argumento [0] = nombre del archivo python ##
22 ## -argumento [1] = catalogo dado por sextractor con todas las fuentes ##
23 ## fuentes encontradas en la imagen fits correspondiente. ##
24 ## -argumento [2] = datos en columnas de la minima comun region ##
25 ## rectangular ##
26 ## Retorna: ##
27 ## -argumento [3] = region circular de cada imagen ##
28 ## -argumento [4] = coordenadas en grados de la declinacion, ##
29 ## ascension recta de fuentes ##
30 ## dentro de la minima comun region ##
31 ## -argumento [5] = define la cabecera de cada archivo de salida del ##
32 ## argumento [4] ##
33 ## ##
34 #####
35
36 import sys
37 import numpy as np
38
39 #-----
40 ### Argumentos
41 #-----
42
43 # Nombre del archivo a leer, de ahi el [1].Este archivo guarda las
44 # declinaciones y ascenciones rectas obtenidas con sextractor.
45 file=sys.argv[1]
46 # Minima Region rectangular
47 fileRegCuadMini=sys.argv[2]

```

```

48
49 # Nombre del archivo de salida, con las regiones circulares
50 outFile = sys.argv[3]
51 # Nombre del archivo de salida de los topcat, con solo aquellos datos que
52 # est n en el rect ngulo
53 outTopcat = sys.argv[4]
54 # Numero de etiqueta
55 numberEtiq = sys.argv[5]
56
57 #-----
58 ### Extrae informacion.
59 #-----
60
61 # Selecciono los datos y guardo un array con ellos.
62 n,x,y,ra,dec,fAp,f_errAp,flag=np.loadtxt(file,comments='#',unpack=True)
63 nn=n.size
64
65 # datos minima region comun
66 ARmax,ARmin,Decmax,Decmin= np.loadtxt(fileRegCuadMini, comments='#',unpack=True)
67
68 #-----
69 ### Crea los archivos donde guardara las regiones.
70 #-----
71
72 archSalida=open(outFile, "w")
73 archSalidaTopC= open(outTopcat, "w")
74
75 ## Pone la cabecera del archivo topcat
76 archSalidaTopC.write("# N"+str(numberEtiq)+" X"+str(numberEtiq)+" Y"+str(numberEtiq)+
77 " A"+str(numberEtiq)+" D"+str(numberEtiq)+" FAp"+str(numberEtiq)+" F_errAp"+str(
78 numberEtiq)+" F"+str(numberEtiq)+"\n")
79
80 ## Compara y escribe la informacion en los archivos de salida.
81 for i in range(nn):
82     if Decmin<dec[i] and dec[i]<Decmax:
83         if ARmin<ra[i] and ra[i]<ARmax:
84             archSalida.write("fk5;circle("+str(ra[i])+","+str(dec[i])+',,2") # color=
85             green text={' +str(n[i])+'}; \n')
86             archSalidaTopC.write(str(int(n[i]))+" "+str(x[i])+ " "+str(y[i])+ " "+str(
87 ra[i])+ " "+str(dec[i])+ " "+str(fAp[i])+ " "+str(f_errAp[i])+ " "+str(int(flag[i]))+"
88 \n")
89
90 archSalida.close()
91 archSalidaTopC.close()
92
93 ## Fin marcaCircFuentes.py -----

```

```

1 #!/usr/bin/env python2
2 # -*- coding: utf-8 -*-
3 """
4 Created on April 2019
5 @author: Maria Ruiz Perez
6 """
7 #####
8 ##                                PR5                                Nombre: findingCharts.py                                ##
9 ##-----##
10 ## Programa para leer el archivo conseguido con stilts y retornar un                                ##
11 ## fichero para ser entendido por ds9.                                ##
12 ##                                ##
13 ## Requiere: archivo conseguido previamente con stilts.                                ##
14 ## Retorna:  archivo con las instrucciones para ser entendidas en ds9                                ##
15 ##                                y generar las finding charts.                                ##
16 #####
17
18
19 import sys
20 import numpy as np
21
22
23 # Nombre del archivo a leer, tmatchn
24 archTMATCHN = sys.argv[1]
25 # Nombre archivo salida DS9
26 archDS9 = sys.argv[2]
27 #-----
28 ### Selecciona los datos y guarda un array con ellos.
29 #-----
30 # Se han a adido 3 variables, dos para leer los datos de la imagen os_median
31 # A1 y D1; y la variable indice para leer la columna de los indices a adidos
32 # en otro archivo python.
33 A1,D1,indice= np.genfromtxt(archTMATCHN,comments='#',unpack=True,usecols=(3,4,56),
34                             dtype='str')
35
36
37
38 #-----
39 ### Crea un archivo donde guarda las instrucciones para ds9.
40 #-----
41
42 direcOUT = open(archDS9, "w")
43
44 ## Comparo.
45 for i in range(dimens):
46     # Cada frame se asocia con una min. comun reg.de nombre:
47     #         -region load regCuadMin.reg
48     # Se ha cambiado el nombre de las imagenes jpg. Con zfills(number) a ades
49     # a un string ceros a la izquierda.
50     # Se ha a adido un frame mas para tener en cuenta la nueva imagen.
51     # Para entender mejor se ha cambiado el nombre de las regiones circulares
52     # antes eran m***.reg y ahora son m***Circ.reg
53     direcOUT.write("ds9 m0s_8.fits m736_1.fits m738_2.fits m739_3.fits m740_4.fits
54 m741_5.fits m742_6.fits m744_7.fits -geometry 1000x1000 -frame 1 -wcs align yes -
55 pan to "+ A1[i]+" "+D1[i]+" wcs fk5 -match frames wcs -frame 1 -invert -region
56 load m736Circ_1.reg -region load regionComun.reg -scale linear -scale mode zscale
57 -scale scope local -zoom to 2 -frame 2 -invert -region load m736Circ_1.reg -region

```

```

    load regionComun.reg -scale linear -scale mode zscale -scale scope local -zoom to
    2 -frame 3 -invert -region load m738Circ_2.reg -region load regionComun.reg -
    scale linear -scale mode zscale -scale scope local -zoom to 2 -frame 4 -invert -
    region load m739Circ_3.reg -region load regionComun.reg -scale linear -scale mode
    zscale -scale scope local -zoom to 2 -frame 5 -invert -region load m740Circ_4.reg
    -region load regionComun.reg -scale linear -scale mode zscale -scale scope local -
    zoom to 2 -frame 6 -invert -region load m741Circ_5.reg -region load regionComun.
    reg -scale linear -scale mode zscale -scale scope local -zoom to 2 -frame 7 -
    invert -region load m742Circ_6.reg -region load regionComun.reg -scale linear -
    scale mode zscale -scale scope local -zoom to 2 -frame 8 -invert -region load
    m744Circ_7.reg -region load regionComun.reg -scale linear -scale mode zscale -
    scale scope local -zoom to 2 -view colorbar no -saveimage jpg "+"/home/maria/
    Escritorio/TFGcodigos/consola/fotosDS9/foto"+indice[i].zfill(4)+".jpg -quit \n\n")
52
53
54 direcOUT.close()
55
56 ## Fin findingCharts.py -----
57
58 1 #!/usr/bin/env python3
59 2 # -*- coding: utf-8 -*-
60 3 """
61 4 Created on April 2019
62 5 @author: Maria Ruiz Perez
63 6 """
64 7 #####
65 8 ##          PR5.1          Nombre: generaFindingCharts.py          ##
66 9 ##-----##
70 10 ## Programa para leer el archivo conseguido con stilts y retornar un          ##
71 11 ## fichero para ser entendido por ds9.          ##
72 12 ## Requiere: archivo conseguido previamente con stilts.          ##
73 13 ## Retorna: archivo con las instrucciones para ser entendidas en ds9          ##
74 14 ##          y generar las finding charts.          ##
75 15 #####
76 16
77 17
78 18 import sys
79 19 import numpy as np
80 20
81 21
82 22 # Nombre del archivo a leer, tmatchn
83 23 archTMATCHN = sys.argv[1]
84 24 # Nombre archivo salida DS9
85 25 archDS9 = sys.argv[2]
86 26 #-----
87 27 ### Selecciona los datos y guarda un array con ellos.
88 28 #-----
89 29 # Se han a adido 3 variables, dos para leer los datos de la imagen os_median
90 30 # A1 y D1; y la variable indice para leer la columna de los indices a adidos
91 31 # en otro archivo python.
92 32 A1,D1,indice= np.genfromtxt(archTMATCHN,comments='#',unpack=True,usecols=(3,4,56),
93                             dtype='str')
94 33 dimens = A1.size
95 34
96 35 #-----
97 36 ### Crea un archivo donde guarda las instrucciones para ds9.
98 37 #-----

```

```

38
39 direcOUT = open(archDS9, "w")
40
41 ## Comparo.
42 for i in range(dimens):
43     # Cada frame se asocia con una min. comun reg.de nombre:
44     #         -region load regCuadMin.reg
45     # Se ha cambiado el nombre de las imagenes jpg. Con zfills(number) a ades
46     # a un string ceros a la izquierda.
47     # Se ha a adido un frame mas para tener en cuenta la nueva imagen.
48     # Para entender mejor se ha cambiado el nombre de las regiones circulares
49     # antes eran m***.reg y ahora son m***Circ.reg
50     direcOUT.write("ds9 m0s_8.fits m736_1.fits m738_2.fits m739_3.fits m740_4.fits
m741_5.fits m742_6.fits m744_7.fits -geometry 1000x1000 -frame 1 -wcs align yes -
pan to "+ A1[i]+" "+D1[i]+" wcs fk5 -match frames wcs -frame 1 -invert -region
load m736Circ_1.reg -region load regionComun.reg -scale linear -scale mode zscale
-scale scope local -zoom to 2 -frame 2 -invert -region load m736Circ_1.reg -region
load regionComun.reg -scale linear -scale mode zscale -scale scope local -zoom to
2 -frame 3 -invert -region load m738Circ_2.reg -region load regionComun.reg -
scale linear -scale mode zscale -scale scope local -zoom to 2 -frame 4 -invert -
region load m739Circ_3.reg -region load regionComun.reg -scale linear -scale mode
zscale -scale scope local -zoom to 2 -frame 5 -invert -region load m740Circ_4.reg
-region load regionComun.reg -scale linear -scale mode zscale -scale scope local -
zoom to 2 -frame 6 -invert -region load m741Circ_5.reg -region load regionComun.
reg -scale linear -scale mode zscale -scale scope local -zoom to 2 -frame 7 -
invert -region load m742Circ_6.reg -region load regionComun.reg -scale linear -
scale mode zscale -scale scope local -zoom to 2 -frame 8 -invert -region load
m744Circ_7.reg -region load regionComun.reg -scale linear -scale mode zscale -
scale scope local -zoom to 2 -view colorbar no -saveimage jpg "+/home/maria/
Escritorio/TFGcodigos/consola/fotosDS9/foto"+indice[i].zfill(4)+".jpg -quit \n\n")
51
52
53 direcOUT.close()
54
55 ## Fin generaFindingCharts.py -----

```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 """
5 Created on April 2019
6 @author: Maria Ruiz Perez
7 """
8
9 #####
10 ## PRF2                MAIN 2- FASE 2: FOTOMETRIA Y ANALISIS                ##
11 ##                                DE INFORMACION EXTRAIDA                ##
12 ##-----##
13 ##      Archivo con el que se realiza toda la fase 2. Registro de los scripts ##
14 ##      creados por el orden en que deben ser ejecutados en la terminal.    ##
15 #####
16
17
18 """
19 Invoca PR 6
20 Nombre:  calibracionConEstrella.py
21 Requiere:
22     -Espectro de la estrella de calibracion (incluido en el codigo programa)
23     -Fichero de datos de los filtros (incluido en el codigo del programa)
24 Retorna:
25     -Fichero con parametros principales calculados de calibracion por cada imagen.
26     -Fichero con los flujos y sus errores empleando el factor de conversion.
27     -Representaciones.
28 """
29
30 python calibracionConEstrella.py
31
32
33 """
34 Invoca PR 7
35 Nombre:  suavizadoEspectro.py
36 Requiere:
37     -Espectro del cuasar.
38 Retorna:
39     -Grafico con espectro suavizado sobre espectro sin suavizar.
40 """
41
42 python suavizadoEspectro.py
43
44 """
45 Invoca PR 8
46 Nombre:  ajusteChiMideZ.py
47 Requiere:
48     -Espectro del cuasar.
49 Retorna:
50     -Grafico de ajuste minimizando a chi cuadrado del espectro
51     observado sin suavizar sobre el espectro observado sin suavizar.
52 """
53
54 ## FIN allMainFase2.sh -----

```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on 24 May 2019.
5 @author: Maria Ruiz Perez
6 Unidades en el sistema cgs
7 """
8
9 #####
10 ##                               PR6                               Nombre: calibracionConEstrella.py                               ##
11 ##-----##
12 ## Programa destinado a la calibracion de las imagenes astronomicas                               ##
13 ## originales .fits tomadas con el telescopio. Con otros codigos se han                               ##
14 ## detectado las fuentes en las imagenes, dando lugar a catalogos a partir de                               ##
15 ## los cuales se han conseguido flujos. Este programa calibra estos flujos                               ##
16 ## ademas de generar graficos y archivos con resultados.                               ##
17 ## La calibracion se realiza a partir de una estrella en la region de cielo                               ##
18 ## que se esta estudiando.                               ##
19 ## -Coordenadas de la estrella: AR=119.04307grad De=41.08158grad                               ##
20 #####
21
22 import numpy as np
23 import matplotlib.pyplot as plt
24 import matplotlib as mpl
25 from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, AutoMinorLocator)
26 from astropy.io import fits as ft
27
28
29 #-----
30 ### Funcion para integrar. Emplea el metodo trapezoidal simple.
31 ### Parametros: x,f ,deben ser arrays de 1Dim.
32 #-----
33 def intTrap(x,f):
34     # Longitud del arrayND
35     dim= f.size
36     # Calculo todas las integrales en tramos sucesivos
37     integral = (x[1:]-x[:dim-1])*(f[1:]+f[:dim-1])/2
38     # Calculo la integral total
39     integral = np.sum(integral)
40     return integral
41
42
43 #-----
44 ### Extraccion de informacion.
45 #-----
46
47 ## Fichero de datos de los filtros.
48 nameFile = "Lambda_Width_0009_J0756.topcat"
49 # Datos del fichero
50 informacionFiltros=np.loadtxt(nameFile)
51
52
53 ##### Variables de trabajo.
54 # Longitudes de onda de los filtros.
55 lambda0 = informacionFiltros[:,0]
56 # Paso el sistema cgs, en este caso cm

```

```

57 lambdas_cm=lambda0*1e-8
58 # Ancho de los filtros
59 width = informacionFiltros[:,1]
60 # Paso el sistema cgs, en este caso cm
61 width = width*1e-8
62
63
64
65 # Espectro de la estrella
66 informacionEspectro=ft.open("spec-3680-55210-0370.fits")
67 # Flujo especifico obtenida del espectro, dE/dtdSdlLambda
68 # Unidades del flujo especifico: erg/(sec. cm^2 cm)
69 # Datos tipo array nD
70 flujo = informacionEspectro[1].data["flux"]
71 # Las longitudes de onda en base log 10.
72 loglambda = informacionEspectro[1].data["loglam"]
73 # Quito el logaritmo en base 10
74 lambdadaa=np.float_power(10,loglambda)
75 # Hace la conversion de los datos SDSS a los valores equivalentes en STP
76 lambdadaa= lambdadaa / (1.0 + 2.735182e-4 + 131.4182/lambdadaa**2 + 2.76249e8/lambdadaa**4)
77 # Expreso las longit. de onda del espectro en cm
78 lambdadaa = lambdadaa*1e-8
79 #Error del flujo del espectro
80
81 # Longitud de los arrays utilizados
82 dimLamEsp = lambdadaa.size
83 dimLam0 = lambda0.size
84
85
86
87 #-----
88 ### Calcula la transmision, T(lambda)
89 #-----
90 # Cada casilla sera la transmision del cuasar.
91 # Hay una funcion de transmision por imagen:
92 # para la Imagen 1 T1, para la imagen 2,T2 ...
93 # T equivale a tener una matriz, donde T1 esta en la fila 1, T2 en la fila 2,...
94
95 T = np.zeros([dimLam0,dimLamEsp])
96 # Cambia de filtro, (lambda0_cm, width)
97 for i in range(dimLam0):
98     # Cambia las lognitudes de onda del espectro (lambdadaa)
99     for j in range(dimLamEsp):
100         denomT = 1+(4*(lambdadaa[j]-lambdas_cm[i])**2)/width[i]**2
101         T[i,j] = 1/denomT
102
103
104 if True:
105
106 #-----
107 ### Calcula el Flujo esperado.
108 #-----
109 # Se tiene un flujo esperado por imagen, es decir un flujo
110 # por cada lambda0. Unidades: dE/(dtdSdlLambda)
111
112 flujoEsperado = np.zeros(dimLam0)

```

```

113     for i in range(dimLam0):
114         # (numerador) Integrando flujo*transmision
115         funcion1 = flujo*T[i,:]
116         # Integral flujo*transmision
117         numerador = intTrap(lambdaa,funcion1)
118         # Integral denominador:
119         denominador = intTrap(lambdaa,T[i,:])
120         flujoEsperado[i] = numerador/denominador
121
122
123     #-----
124     ### Busco la estrella en el archivo que alberga todos los flujos de cada
125     ### una de las fuentes de la regi n com n de entre todas las im gines
126     ### originales:
127     #-----
128
129     # Archivo de combinacion obtenido durante la fase 1
130     nameFile = "newOS_combinacionTopcatV2_match.topcat"
131     # Columnas asR1,Dec1, Flux1,FluxEr1,Flux2,FluxEr2,Flux3,FluxEr3,...
132     colum = (3,4,5,6,13,14,21,22,29,30,37,38,45,46,53,54,56)
133     A1,D1,F1,FE1,F2,FE2,F3,FE3,F4,FE4,F5,FE5,F6,FE6,F7,FE7,index=np.genfromtxt(nameFile
134         ,comments='#',unpack=True,usecols=colum)
135
136     # Coordenadas de la estrella: (Obtenidas del SDSS)
137     arStar = 119.04307
138     decStar = 41.08158
139
140     # Las cuentas las guardare en este arrayND.
141     C_j = np.zeros(dimLam0)
142
143     # Flujos de la estrella
144     flujStar = flujoEsperado
145     # Errores de los flujos de la estrella (en cuentas)
146     ErrorflujStar = np.zeros(dimLam0)
147
148     # Lista de estrellas posibles.
149     for i in range(len(A1)):
150         # En funcion de las condiciones impuestas se encuentran mas o menos
151         # posibles cuasares. Con las cond. fijadas ahora se encuentra solo uno.
152         if abs(A1[i]-arStar)<1e-2 and abs(D1[i]-decStar)<1e-4:
153             # Observo posibles cuasares.
154             #print(str(A1[i])+" "+str(D1[i]))
155             print('indice de la estrella: '+str(i+1))
156             # Guardo las cuentas (flujos) del quasar; Las cuentas C_j se denotan en el
157             # archivo de salida del tmachn como FAp
158             C_j[0]=F1[i]
159             ErrorflujStar[0]=FE1[i]
160             C_j[1]=F2[i]
161             ErrorflujStar[1]=FE2[i]
162             C_j[2]=F3[i]
163             ErrorflujStar[2]=FE3[i]
164             C_j[3]=F4[i]
165             ErrorflujStar[3]=FE4[i]
166             C_j[4]=F5[i]
167             ErrorflujStar[4]=FE5[i]
168             C_j[5]=F6[i]

```

```

167         ErrorflujStar[5]=FE6[i]
168         C_j[6]=F7[i]
169         ErrorflujStar[6]=FE7[i]
170         print(C_j)
171
172     ## Calcula del factor de conversion de cuentas a flujos.
173     x_i = flujoEsperado/C_j
174     print(x_i)
175
176     ## Convierte el flujo del cuasar y los errores del flujo del cuasar en cuentas, a
177     unidades de flujos.
178     ErrorStar=ErrorflujStar*x_i
179     flujStar=C_j*x_i
180
181     ## Calcula los flujos y su error de todas las fuentes de cada una de las fotos
182     empleando el factor de conversion anterior.
183     fluj = np.array([F1*x_i[0],F2*x_i[1],F3*x_i[2],F4*x_i[3],F5*x_i[4],F6*x_i[5],F7*x_i
184     [6]])
185     flujError = np.array([FE1*x_i[0],FE2*x_i[1],FE3*x_i[2],FE4*x_i[3],FE5*x_i[4],FE6*
186     x_i[5],FE7*x_i[6]])
187
188     #-----
189     ### Fichero #1 de salida. Escribe los parametros calculados por
190     ### imagen en un fichero
191     #-----
192     nFileOut = "Parametros_estrella"
193     # Parametros por foto. Escribe sobre el fichero de salida.
194     archOut = open(nFileOut+".cat","w")
195     # Escribe la cabecera del archivo de salida
196     archOut.write("# flujEspec"+"      CountsStar"+"      FactConv"+"\\n")
197     for j in range(dimLam0):
198         archOut.write(str(flujoEsperado[j])+"      "+str(C_j[j])+"      "+str(x_i[j])+"\\n")
199
200     archOut.close()
201
202     #-----
203     ### Fichero #2 de salida. Escribe los flujos y sus errores empleando
204     ### el factor de conveersion.
205     #-----
206     nombre = "flujosCalibrados_estrella.topcat"
207     archOutFluj = open(nombre,"w")
208     # Escribe la cabecera del archivo de salida
209     archOutFluj.write("#Index"+"      "+AscRecta+"      "+Declinac+"      "+flujo1+"      "+
210     flujError1+"      "+flujo2+"      "+flujoError2+"      "+flujo3+"      "+flujoError3"
211     +"      "+flujo4+"      "+flujoError4+"      "+flujo5+"      "+flujoError5+"      "+
212     flujError6+"      "+flujoError6+"      "+flujo7+"      "+flujoError7+"\\n")
213     # Escribe en un solo archivo de salida todos los flujos calibrados.
214     for j in range(len(fluj[0,:])):
215         archOutFluj.write(str(int(index[j]))+"      "+str(A1[j])+"      "+str(D1[j])+"
216         "+str(fluj[0,j])+"      "+str(flujError[0,j])+"      "+str(fluj[1,j])+"      "+str(
217         flujError[1,j])+"      "+str(fluj[2,j])+"      "+str(flujError[2,j])+"      "+str(fluj[3,j]
218         )+"      "+str(flujError[3,j])+"      "+str(fluj[4,j])+"      "+str(flujError[4,j])+"      "+
219         str(fluj[5,j])+"      "+str(flujError[5,j])+"      "+str(fluj[6,j])+"      "+str(flujError
220         [6,j])+"\\n")
221
222     archOutFluj.close()

```

```

211
212 #####
213 # REPRESENTACIONES #
214 #####
215
216 # Ruta donde guarda las figuras que a continuaci n realiza.
217 rutaSalida = "/home/maria/Escritorio/TFGcodigos/consola/"
218 # Formato en el que se van a guardar todas las gr ficas que el
219 # programa realiza.
220 formato = "jpg"
221
222 #-----
223 # Representacion del factor de transmision. Adimensional.
224 #-----
225 labels = []
226 fig1 = plt.figure(1)
227 for i in range(dimLam0):
228     # Convierte longitudes de onda a nm.
229     plt.plot(lambdaa*1e7,T[i,:],color='b')
230     longOnda = str(round(lambdas_cm[i]*1e7,2))
231     #round redondea a dos decimales
232     labels.append(r'$\lambda = $'+longOnda)
233
234 axes = plt.gca()
235 axes.set_xlim([730,750])
236 axes.minorticks_on()
237 axes.tick_params(axis = 'x', which='minor', direction='out',labelsize = 4)
238 plt.legend(labels)
239
240 # A ade el titulo.
241 # Etiquetas en los ejes.
242 plt.xlabel(r'$\lambda$ (nm)$')
243 plt.ylabel(r'$T(\lambda)$')
244 # Guarda la figura.
245 plt.savefig(rutaSalida+"Transmisi n de filtros vs lambdas filtros",format=formato,
246             box_inches='tight')
247 plt.show()
248 #-----
249 ### Representacion del flujo observado.
250 #-----
251
252 fig0 = plt.figure(2)
253
254 # Pone ticks a todos los ejes
255 plt.rc('xtick',top=True)
256 plt.rc('ytick',right=True)
257 # Ejes
258 axes=plt.axes()
259 # Pone ticks menores a los ejes.
260 axes.xaxis.set_minor_locator(MultipleLocator(20))
261 axes.yaxis.set_minor_locator(MultipleLocator(10))
262 # Convierte long. de onda a nm
263 plt.plot(lambdaa*1e7,flujo)
264
265 # Etiquetas ejes.

```

```

266 plt.ylabel(r'$F_{\lambda} (10^{-17}\mathrm{erg} \text{ cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1})$')
267 plt.xlabel(r'$\lambda (\mathrm{nm})$')
268 plt.xlim(340,1040)
269 plt.ylim(0,260)
270
271 # Añade rayas verticales que marcan el rango espectral abarcado por los filtros en
    su conjunto
272 plt.vlines(732,0,50,color='red')
273 plt.vlines(748,0,50,color='red')
274 # Añade leyenda.
275 leyenda=['Flujo observado','Rango de interes']
276 plt.legend(leyenda,loc='upper right',bbox_to_anchor=(1,1))
277
278 # Guarda la figura
279 plt.savefig(rutaSalida+"Flujo observado vs lambda",format=formato, bbox_inches='tight')
280
281 # Muestra la figura en pantalla
282 plt.show()
283
284 ##### Representacion del flujo observado en un zoom de solo el
285 ##### rango espectral que abarcan los filtros.
286 #####
287 fig5 = plt.figure(3)
288
289 # Pone ticks a todos los ejes.
290 plt.rc('xtick',top=True)
291 plt.rc('ytick',right=True)
292 # Ejes.
293 axes=plt.axes()
294 # Pone ticks menores a los ejes.
295 axes.xaxis.set_minor_locator(MultipleLocator(0.5))
296 axes.yaxis.set_minor_locator(MultipleLocator(2))
297 # Convierto long. de onda a nm
298 plt.plot(lambd*1e7,flujo)
299 # Etiquetas ejes.
300 plt.ylabel(r'$F_{\lambda} (10^{-17}\mathrm{erg} \text{ cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1})$')
301 plt.xlabel(r'$\lambda (\mathrm{nm})$')
302 plt.xlim(730,750)
303 plt.ylim(0,55)
304 # Añadir rayas verticales que marcan el rango espectral abarcado por los
305 # filtros en su conjunto
306 plt.vlines(732,0,40,color='red')
307 plt.vlines(748,0,40,color='red')
308 # Leyenda.
309 leyenda=['Flujo observado','Rango de interes']
310 plt.legend(leyenda,loc='upper right',bbox_to_anchor=(1,1))
311 # Guarda la figura
312 plt.savefig(rutaSalida+"Flujo observado en ZOOM vs lambda",format=formato,
    bbox_inches='tight')
313 # Muestra la figura en pantalla
314 plt.show()
315
316 #####
317 ##### Representacion del flujo del cuasar solo.
318 #####

```

```

319
320 fig2=plt.figure(4)
321 plt.errorbar(lambdas_cm*10**7,flujStar,yerr=abs(ErrorflujStar),marker=".",linestyle="
    None")
322 # Etiquetas.
323 plt.ylabel(r'$F_{10^{-17}\mathrm{erg}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}}\AA^{-1}$')
324 plt.xlabel(r'$\lambda_{\mathrm{nm}}$')
325 plt.xlim(730,750)
326 # Pone ticks a todos los ejes
327 plt.rc('xtick',top=True)
328 plt.rc('ytick',right=True)
329 # Guarda la figura
330 plt.savefig(rutaSalida+"Flujo de la estrella solo vs lambda",format=formato,
    bbox_inches='tight')
331 # Muestra la figura en pantalla
332 plt.show()
333
334 #-----
335 ### Representacion conjunta del flujo del cuasar
336 ### solo mas el flujo observado.
337 #-----
338 fig3 = plt.figure(5)
339 marcad = np.array(['o', 'o', 'o', 'o', 'o', 'o', 'o'])
340 # Convierto long. de onda a nm
341 plt.plot(lambdaa*1e7,flujo)
342 labelNew = ["Flujo observado","puntos"]
343 for k in range(dimLam0):
344     plt.errorbar(lambdas_cm[k]*10**7,flujStar[k],yerr=abs(ErrorflujStar[k]),marker=
        marcad[k],color="red")
345     longOnda = str(round(lambdas_cm[i]*1e7,2))
346
347 plt.legend(labelNew,loc="lower right",bbox_to_anchor=(0.45,0.2))
348
349 # Etiquetas de los ejes
350 plt.ylabel(r'$F_{\lambda_{10^{-17}\mathrm{erg}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}}\AA^{-1}}$')
351 plt.xlabel(r'$\lambda_{\mathrm{nm}}$')
352 # Ajusta los ejes para que se vea mejor la imagen
353 axes = plt.gca()
354 axes.set_xlim([734,746])
355 plt.ylim(17,22)
356 #Pone ticks a todos los ejes
357 plt.rc('xtick',top=True)
358 plt.rc('ytick',right=True)
359 # Guarda la figura
360 plt.savefig(rutaSalida+"Flujo de la estrella y flujo observado vs lambda",format=
    formato, bbox_inches='tight')
361 # Muestra la figura en pantalla
362 plt.show()
363
364
365 #-----
366 ### Representacion del factor de transmision y el flujo observado.
367 #-----
368 fig4 = plt.figure(6)
369 for i in range(dimLam0):
370     # Convierte long. de onda a nm.

```

```

371     longOnda = str(round(lambdas_cm[i]*1e7,2))
372     #labelNew2=r'$\lambda = $'+longOnda
373     plt.plot(lambdaa*1e7,10*T[i,:])
374     plt.text(lambdas_cm[i]*1e7,10.5,str(i+1)) # a ade a cada filtro
375     # un n mero. Cambia su posicion en el eje y, con el segundo argumento.
376
377 # Convierte longitudes de onda a nm.
378 plt.plot(lambdaa*1e7,flujo[:,],label='Flujo obs.')
379 # Dibuja los ticks en todos los ejes.
380 plt.rc('xtick',top=True)
381 plt.rc('ytick',right=True)
382 # Ejes.
383 axes=plt.axes()
384 # Pone ticks menores a los ejes.
385 axes.xaxis.set_minor_locator(MultipleLocator(0.5))
386 axes.yaxis.set_minor_locator(MultipleLocator(1))
387 # Ajusta los ejes para ver mejor la imagen.
388 plt.xlabel(r'$\lambda$ (\mathrm{nm})$')
389 plt.ylabel(r'$F_{\lambda}(10^{-17}\mathrm{erg\ cm^{-2}\ s^{-1}\ \AA^{-1}})$')
390 # Pone ticks a todos los ejes.
391 plt.rc('xtick',top=True)
392 plt.rc('ytick',right=True)
393 # Grafica.
394 plt.errorbar(lambdas_cm*1e7,flujStar,yerr=ErrorflujStar,color="red",ecolor="b",fmt='.',
395             ',marker="o",ms=3,linestyle="None",label="Flujo calib.")
396
397 # Ajusta la leyenda a la posici n que se desee.
398 plt.legend(loc="lower right",bbox_to_anchor=(0.28,0.15))
399 plt.xlim(730,750)
400 plt.ylim(0,24)
401 # Guarda la figura
402 plt.savefig(rutaSalida+"Flujo observado.jpg",format=formato, bbox_inches='tight')
403 # Muestra la figura en pantalla
404 plt.show()
405 ## FIN calibracionConEstrella -----

```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 """
5 Fecha: 05/05/19.
6 Autor: Maria Ruiz
7 Unidades en el sistema cgs
8 """
9 #####
10 ##          PR 7          Nombre:  suavizadoEspectro.py          ##
11 ##-----##
12 ## Programa que calcula el flujo observado del cuasar suavizado por un filtro      ##
13 ## de ancho 13.6 nm y representa este flujo sobre el flujo observado del cuasar.  ##
14 ##                                                                                      ##
15 ## Requiere:                                                                                      ##
16 ##          -Espectro del cuasar.                                                                                      ##
17 ## Retorna:                                                                                      ##
18 ##          -Grafico con espectro suavizado sobre espectro sin suavizar.          ##
19 ##                                                                                      ##
20 #####
21 import numpy as np
22 import matplotlib.pyplot as plt
23 import matplotlib as mpl
24 from matplotlib.ticker import MultipleLocator, FormatStrFormatter,
25                               AutoMinorLocator
26 from astropy.io import fits as ft
27
28
29 #-----
30 ### Filtros.
31 #-----
32
33 ## Archivo de filtros.
34 nameFile = "Lambda_Width_0009_J0756.topcat"
35 ## Informacion de filtros.
36 informacionFiltros=np.loadtxt(nameFile)
37
38 ## Variables de trabajo:
39 # Longitudes de onda de los filtros.
40 lambda0 = informacionFiltros[:,0]
41 # Pasa el sistema cgs, en este caso cm
42 lambdas_cm=np.float_power(10,-8)*lambda0
43 # Ancho de los filtros
44 width = informacionFiltros[:,1]
45 # Pass the cgs system, in this case cm
46 width = width*1e-8
47
48 #-----
49 ### Cuasar.
50 #-----
51
52 # Espectro del cuasar
53 informacionEspectro=ft.open("spec-cuasar-3680-55210-0366.fits")
54 # Flujo especifico obtenida del espectro,dE/dtdSdlLambda
55 # Unidades del flujo especifico: erg/(sec. cm^2 cm)
56 # Datos tipo array nD

```

```

57 flujo = informacionEspectro[1].data["flux"]
58 # Las longitudes de onda en base log 10.
59 loglambda = informacionEspectro[1].data["loglam"]
60 # Quito el logaritmo en base 10
61 lambdaa=np.float_power(10,loglambda)
62 # Hace la conversion de los datos SDSS a los valores equivalentes en STP
63 lambdaa= lambdaa / (1.0 + 2.735182e-4 + 131.4182/lambdaa**2 + 2.76249e8/lambdaa**4)
64 # Expreso las longit. de onda del espectro en cm
65 lambdaa = lambdaa*(10**-8)
66 #Error del flujo del espectro
67
68 # Longitud de los arrays utilizados
69 dimLamEsp = lambdaa.size
70 dimLam0 = lambda0.size
71
72 #####
73 #####          CALCULOS          #####
74 #####
75
76 """
77 Calculo de una integral empleando el metodo trapezoidal simple.
78 Parametros: x,f ,deben ser arrays 1Dim.
79 """
80
81 def intTrap(x,f):
82     # longitud del arrayND
83     dim= f.size
84     # Calculo todas las integrales en tramos sucesivos
85     integral = (x[1:]-x[::(dim-1)])*(f[1:]+f[::(dim-1)])/2
86     # Calculo la integral total
87     integral = np.sum(integral)
88     return integral
89
90 #-----
91 ### Calcula de la transmision, T(lambda)
92 #-----
93 T = np.zeros([dimLamEsp,dimLamEsp])
94 # El factor T es el de un solo filtro, luego solo se necesita que sea una fila.
95
96 # lambdaa es todo el espectro de long de onda.
97 # lamdabs_cm solo me interesa el del filtro que tiene ancho 13.6, que es el de la
98 # casilla numero 1
99 for i in range(dimLamEsp):
100     for j in range(dimLamEsp):
101         denomT = 1+(4*(lambdaa[j]-lambdaa[i])**2)/width[1]**2
102         T[i,j] = 1/denomT
103 #-----
104 ### Calcula el flujo a traves del filtro.
105 #-----
106 flujoEsperado=np.zeros(dimLamEsp)
107 for i in range(dimLamEsp):
108     #(numerador) Integrando flujo*transmision
109     funcion1=flujo*T[i,:]
110     #Integral flujo*transmision
111     numerador=intTrap(lambdaa,funcion1)
112     #Integral denominador:

```

```

113     denominador=intTrap(lambdaa,T[i,:])
114     flujoEsperado[i]=numerador/denominador
115
116 #####
117 ##### REPRESENTACION #####
118 #####
119
120 # Representa el flujo observado junto con el flujo observado suavizado.
121
122 plt.figure(1)
123 # Pone ticks en todos los ejes
124 plt.rc('xtick',top=True)
125 plt.rc('ytick',right=True)
126 # Ejes.
127 axes=plt.axes()
128 # Pone ticks menores a los ejes.
129 axes.xaxis.set_minor_locator(MultipleLocator(4))
130 axes.yaxis.set_minor_locator(MultipleLocator(0.2))
131 # Dibuja
132 plt.plot(lambdaa*1e7,flujo, color='black',label='Flujo observado')
133 plt.plot(lambdaa*1e7,flujoEsperado, color='chartreuse',label='Flujo suavizado')
134 # Modifica los ejes para centrarse en el rango de inter s.
135 plt.xlim(680,840)
136 plt.ylim(-1,5)
137 plt.ylabel(r'$F_{\lambda} (10^{-17}\mathrm{erg} \mathrm{cm}^{-2} \mathrm{s}^{-1} \mathrm{AA}^{-1})$')
138 plt.xlabel(r'$\lambda (\mathrm{nm})$')
139 #-----
140 ### Flujo 0.4 erg cm-2 s-1 AA-1
141 #-----
142 # Linea discontinua de 0.4
143 linDiscX = np.array([680,840])
144 linDiscY = np.array([0.4, 0.4])
145 plt.plot(linDiscX,linDiscY,linestyle='--', color='deepskyblue',label= r'$0.4 \times 10^{-17} \mathrm{erg} \mathrm{cm}^{-2} \mathrm{s}^{-1} \mathrm{AA}^{-1}$')
146 # Lineas verticales para marcar el rango en X
147 # del flujo 0.4
148 linVertX1 = np.array([686.15,686.15])
149 linVertX2 = np.array([694.24,694.24])
150 linVertY1 = np.array([0,1])
151 plt.plot(linVertX1,linVertY1,linestyle='-',color='deepskyblue')
152 plt.plot(linVertX2,linVertY1,linestyle='-',color='deepskyblue')
153 #-----
154 ### Flujo 1.4 erg cm-2 s-1 AA-1
155 #-----
156 # Linea discontinua de 1.4
157 linDiscX = np.array([680,840])
158 linDiscY = np.array([1.4, 1.4])
159 plt.plot(linDiscX,linDiscY,linestyle='--',color='red',label=r'$1.4 \times 10^{-17} \mathrm{erg} \mathrm{cm}^{-2} \mathrm{s}^{-1} \mathrm{AA}^{-1}$')
160 # Lineas verticales para marcar el rango en X
161 # del flujo 1.4
162 linVertX3 = np.array([818.71,818.71])
163 linVertX4 = np.array([825.81,825.81])
164 linVertY2 = np.array([0.8,2])
165 plt.plot(linVertX3,linVertY2,linestyle='-',color='red')
166 plt.plot(linVertX4,linVertY2,linestyle='-',color='red')

```



```
167
168 # Leyenda.
169 # leyenda=['Flujo observado','Rango de interes']
170 plt.legend(loc='upper right',bbox_to_anchor=(1,1))
171 #-----
172 ### Guarda la figura.
173 #-----
174 plt.savefig("suavizadoEspectroCuasar"+" .jpg",box_inches='tight')
175 plt.show()
176
177 ## Fin suavizadoEspectro.py -----
```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on June 2019
5
6 @author: Maria Ruiz Perez
7 """
8 #####
9 ##                               PR 8                               Nombre:  ajusteChiMideZ.py                               ##
10 ##-----
11 ## Crea un modelo de ajuste del espectro del cuasar de las lineas LiIV y CIV. ##
12 ## El metodo implementado es una linea recta y dos gaussinas, una por cada linea. ##
13 ## Se obliga a que: ##
14 ## - Los centros de las gaussinas mantengan la relacion entre las longitudes ##
15 ##   de onda esperadas. ##
16 ## - Las anchuras de las gaussinas sean las mismas. ##
17 ## Requiere: ##
18 ##   -Espectro del cuasar. ##
19 ## Retorna: ##
20 ##   -Grafico de ajuste minimizando a chi cuadrado del espectro ##
21 ##   observado sin suavizar sobre el espectro observado sin suavizar. ##
22 #####
23 import numpy as np
24 import matplotlib.pyplot as plt
25 import scipy.optimize as optimization
26 import matplotlib as mpl
27 from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, AutoMinorLocator)
28
29 # Nombre del fichero del espectro.
30 nameEspectra = 'SDSSJ0756+4104_air.qdp'
31
32
33 # Extrae datos del fichero.
34 datos = np.genfromtxt(nameEspectra, skip_header=2)
35 f, c = datos.shape
36
37 # Listas donde guardara informacion.
38 lambdaEspect = []
39 FluxEspect = []
40 FluxEspect_error = []
41
42
43 # Guarda los datos de interes.
44 for i in range(f):
45     # Seleccina el rango de interes.
46     if str(datos[i,2])!="nan" and datos[i,0]<9800 and 8200<datos[i,0]:
47         lambdaEspect.append(datos[i,0])
48         FluxEspect.append(datos[i,1])
49         FluxEspect_error.append(datos[i,2])
50
51 ##-----
52 ### Representa.
53 ##-----
54 # Grafica con las barras de error
55 mpl.rc('xtick', top=True)
56 mpl.rc('ytick', right=True)

```

```

57 plt.errorbar(lambdaEspect,FluxEspect,yerr=FluxEspect_error,fmt='.',color='r',
    markersize=1.5,marker='o')
58 plt.errorbar(lambdaEspect,FluxEspect,yerr=FluxEspect_error,fmt='or',markersize=1.5,
    barsabove='true')
59 plt.plot(lambdaEspect,FluxEspect,'r',markersize=1.5)
60 plt.ylim(-1,4)
61 plt.xlim(8200,9800)
62
63
64 """
65 Define la forma de la funcion que se ajusta al espectro
66 """
67 def func(x,a,b, A1,mu1,sigma1, A2):
68     mu2 = (1548/1397)*mu1
69     sigma2=sigma1
70     return a+b*x + A1*np.exp(-(x-mu1)**2/(sigma1**2)) + A2*np.exp(-(x-mu2)**2/(sigma2
        **2))
71
72
73
74 #-----
75 ### Realiza el ajuste minimizando a chi cuadrado.
76 #-----
77 # Parametros a ajustar: a, b, A1, mu1, sigma1, A2
78 p0=[1.7,0, 0.4,8440,100, 0.6]
79 # Calcula los parametros
80 param , pcov = optimization.curve_fit(func, lambdaEspect, FluxEspect,p0,
    FluxEspect_error)
81 # Convierte los datos de listas a arrays
82
83 FluxEspect_error = np.asarray(FluxEspect_error)
84 FluxEspect = np.asarray(FluxEspect)
85 lambdaEspect = np.asarray(lambdaEspect)
86
87 FluxSilicio = np.zeros(len(FluxEspect))
88
89
90 # Define la funcion ajustada.
91 curvFit = func(lambdaEspect,*param)
92 print('parametros: '+str(param))
93 #-----
94 ### Escribe los par metros ajustados.
95 #-----
96 archOutResult = open("parametrosAjustadosCurveFit.txt","w")
97 # Pone la cabecera del archivo de salida
98 archOutResult.write("a"+"      "+"b"+"      "+"A1"+"      "+"mu1"+"      "+"sigma1"+"      "+"A2"+"\"
    n")
99 # Escribe en un solo archivo de salida.
100 for j in range(1):
101     archOutResult.write(str(int(param[j]))+"      "+(str(param[j+1]))+"      "+str(param[
        j+2]))+"      "+str(param[j+3]))+"      "+str(param[j+4]))+"      "+str(param[j+5]))+"\n")
102
103 archOutResult.close()
104
105 #-----
106 ### Calcula z con la lambda del SiIV que se corresponde con

```

```
107 ### el cuarto parametro de param. Y muestra el valor de Z.
108 #-----
109 z=(param[3]/1397)-1
110 print('El redshift calculado con el SiIV es: z= '+str(z))
111 #-----
112 ### Representa la funcion ajustada.
113 #-----
114 # Pone ticks a todos los ejes.
115 plt.rc('xtick',top=True)
116 plt.rc('ytick',right=True)
117 # Ejes.
118 axes=plt.axes()
119 # Pone ticks menores a los ejes.
120 axes.xaxis.set_minor_locator(MultipleLocator(40))
121 axes.yaxis.set_minor_locator(MultipleLocator(0.2))
122 # Grafica.
123 plt.plot(lambdaEspect,curvFit,'k')
124 plt.ylabel(r'$F_{\lambda} (10^{-17}\mathrm{erg} \mathrm{cm}^{-2} \mathrm{s}^{-1} \mathrm{\AA}^{-1})$')
125 plt.xlabel(r'$\lambda (\mathrm{\AA})$')
126 # Leyenda.
127 leyenda=['Modelo ajuste','Flujo observado']
128 plt.legend(leyenda,loc='upper right',bbox_to_anchor=(1,1))
129 # Guarda la figura
130 rutaSalida="/home/maria/Escritorio/TFGcodigos/consola/"
131 plt.savefig(rutaSalida+"espectroParaMedirZ"+" .jpg", bbox_inches='tight')
132 plt.show()
133
134 ## Fin ajusteChiMideZ-----
```

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 """
5 Created on April 2019
6 @author: Maria Ruiz Perez
7 """
8
9
10 #####
11 ## PRF3          MAIN 3- FASE 3: TRATAMIENTO DE INFORMACION EXTRAIDA Y      ##
12 ##                                     CREACION DE UN METODO DE SELECCION      ##
13 ##-----##
14 ##          Archivo con el que se realiza toda la fase 2. Registro de los scripts  ##
15 ##          creados por el orden en que deben ser ejecutados en la terminal.      ##
16 #####
17
18 """
19 PR 9
20 Nombre: metodoSeleccion.py
21 Programa que selecciona y clasifica de entre todas las fuentes calibradas las que
22 sean de tipo 1, 2, 3 y 4. Ademas representa todas las fuentes calibradas para
23 cada filtro.
24 Requiere:
25     -archivo de flujos calibrados.
26 Retorna:
27     -fichero con todas las galaxias y su informacion, clasificadas en base a
28     una etiqueta en alguno de los tipos o ninguno.
29 """
30
31 python metodoSeleccion.py
32
33 ##-----
34
35
36 """
37 PR 10
38 Nombre: asociaFindingChartsFuentes.py
39 Programa que asocia cada Finding Chart con la grafica de la fuente correspondiente.
40 Requiere:
41     -Finding charts creadas.
42     -Graficas de fuentes calibradas creadas.
43 """
44
45 python asociaFindingChartFuente.py
46
47 ##-----
48
49
50 """
51 PR 11
52 Nombre: diagramasCriteriosFisicos.py
53 Programa en el que se crean diagramas regidos por los criterios fisicos con los que
54 se
55 crea en otro programa el metodo de seleccion.
56 Los diagramas son una explicacion ilustrativa de los criterios fisicos.
```

```
56 """
57
58 python diagramasCriteriosFisicos.py
59
60 ##-----
61
62
63 """
64 PR 12
65 Nombre: histogramas.py
66 Programa que crea histogramas a partir de calculos realizados en el metodo de
67     seleccion.
68 Permite ilustrar los valores numericos obtenidos con los criterios fisicos.
69 Requiere: -archivo de datos para leer y ser representados.
70 """
71
72 python histogramas.py
73
74 ## Fin allMainFase3.sh -----
```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on 24 May 2019.
5 @author: Maria Ruiz Perez
6 """
7 #####
8 ##          PR9          Nombre: metodoSeleccion.py          ##
9 ##-----#
10 ## Programa que selecciona y clasifica de entre todas las fuentes calibradas##
11 ## las que sean de tipo 1, 2, 3 y 4. Ademas representa todas las fuentes      ##
12 ## calibradas para cada filtro.                                             ##
13 ## Requiere:                                                                ##
14 ##          -Archivo de flujos calibrados.                                ##
15 ## Retorna:                                                                ##
16 ##          -Fichero con todas las galaxias y su informacion, clasificadas  ##
17 ##          en base a una etiqueta en alguno de los tipos o ninguno.      ##
18 #####
19
20 import numpy as np
21 import matplotlib.pyplot as plt
22 import matplotlib as mpl
23
24 # Nombre del fichero de flujos calibrados
25 nombre = "flujosCalibrados_estrella.topcat"
26 #
27 datos = np.loadtxt(nombre)
28 # longitudes de onda:
29 longOnd = np.array([736.54,738.54,739.55,740.54,741.55,742.54,744.53])
30 # valores de entrada de los limites de las condiciones
31 # valor beta limite del filtro 2 y 3
32 betaLim23=float(input("Introduzca un valor para que el flujo detectado por el filtro
33     2 y el del filtro 3 sean practicamente igual, e.g menor a 1: "))
34 # valor beta limite del filtro 3 y 4
35 betaLim34=betaLim23
36 # valor alpha limite del filtro 2 y 3
37 alphaLim23=float(input("Introduzca un valor para que el flujo detectado por el filtro
38     2 sea mayor a lo detectado por filtro 3, e.g 2: "))
39 # valor gamma limite del filtro 1 y 2
40 gammaLim21=float(input("Introduzca el cociente m ximo entre el flujo detectado por
41     el filtro 2 entre el del filtro 1, e.g 2.0: "))
42 # valor gamma limite del filtro 1 y 3
43 gammaLim31=gammaLim21
44 A12=float(input("Introduzca el cociente de flujo entre el filtro 2 y filtro 1 del QSD
45     absorbido por los tomos de Hidrogeno, e.g entre 3.5 y 5.5: A= "))
46 print('NOTA: La absorci n de flujo del cociente entre el filtro 2 y filtro 1 del QSD
47     ser la misma para la absorci n del cociente flujo del filtro 3 y filtro 1 del
48     QSD ')
49 A13=A12
50 galaxiasTipo=input(("Introduzca el nombre del archivo de salida que guarda las
51     coordenadas y flujos de todas las galaxias: "))
52 print('-----')
53 print('betaLim23= '+ str(betaLim23))
54 print('alphaLim23= '+str(alphaLim23))
55 print('betaLim34 tiene el mismo valor que tenga betaLim23.')
```

```
50 print('gammaLim31 tiene el mismo valor que ratio21')
51 print('Cociente de flujo entre el filtro 2 y filtro 1 del QSD absorbido por los
      tomos de Hidrogeno es: '+ str(A12))
52
53
54 # Filas y columnas
55 f,c = datos.shape
56
57 print('numero de filas en data: '+str(f))
58
59 # Elimina negativos.
60 zero=np.zeros(datos.shape)
61 datos=np.where(datos<0.,zero,datos)
62 # si se cumple que datos<0 cambia el elemento por cero,
63 # si no se cumple deja el valor del elemento que tenga datos.
64
65 flag = np.zeros(f)
66 formula=np.zeros(f)
67 #-----
68 ### Etiquetado de clasificacion:
69 # Selecciona galaxias.
70 # LAB = 1
71 # LAE = 2
72 # ninguna = 0
73 #-----
74 # LABs encontradas son guardadas en:
75 listaLABS=[]
76 # LAEs encontradas son guardadas en:
77 listaLAES=[]
78
79 # LABs caso1 encontradas son guardadas en:
80 listaLABScaso1=[]
81 # LABs caso2 encontradas son guardadas en:
82 listaLABScaso2=[]
83 # LAEs caso3 encontradas son guardadas en:
84 listaLAEScaso3=[]
85 # LAEs caso4 encontradas son guardadas en:
86 listaLAEScaso4=[]
87
88 #extracting index
89 index=datos[:,0].astype(int)
90 # extracting all fluxes
91 F1=datos[:,3]
92 eF1=datos[:,4]
93 F2=datos[:,5]
94 eF2=datos[:,6]
95 F3=datos[:,7]
96 eF3=datos[:,8]
97 F4=datos[:,9]
98 eF4=datos[:,10]
99 F5=datos[:,11]
100 eF5=datos[:,12]
101 F6=datos[:,13]
102 eF6=datos[:,14]
103 F7=datos[:,15]
104 eF7=datos[:,16]
```



```

105 # ALPHA
106 alpha23=(F2-F3)/np.sqrt(eF2*eF2+eF3*eF3)
107 # calculating differences BETA
108 beta23=np.abs((F2-F3))/np.sqrt(eF2*eF2+eF3*eF3)
109 beta34=np.abs((F3-F4))/np.sqrt(eF3*eF3+eF4*eF4)
110 # calculating the ratios
111 X23=F2/F3
112 eX23=np.sqrt(eF2*eF2/(F3*F3)+(F2*F2)*(eF3*eF3)/np.power(F3,4))
113 X21=F2/F1
114 eX21=np.sqrt(eF2*eF2/(F1*F1)+(F2*F2)*(eF1*eF1)/np.power(F1,4))
115 X31=F3/F1
116 eX31=np.sqrt(eF3*eF3/(F1*F1)+(F3*F3)*(eF1*eF1)/np.power(F1,4))
117 # condicion similar al cuasar GAMMA
118 gamma31=np.abs(X31-A13)/eX31
119 gamma21=np.abs(X21-A12)/eX21
120
121
122
123 print("\n\n Starting processing sources")
124 for i in range(f):
125     print(' Working on i+1 {0:d} index {1:d} '.format(i+1,index[i]))
126     # Selecciono los flujos; flujo es un array de siete casillas
127     flujo = datos[i,3:16:2]
128     # Selecciono los errores de los flujos.; es un array de siete casillas.
129     # Cada una de ellas guarda el error del flujo medido con un filtro.
130     errorFlujo = datos[i,4:17:2]
131     #-----
132     ## Fuente NO detectada en 1, SI en 2, 3 y 4.
133     #-----
134     if flujo[0]<=0 and flujo[1]>0 and flujo[2]>0 and flujo[3]>0:
135         # Si NO esta detec. en 1, SI en 2 y Si en 3.
136         # caso 1: detecci n del filtro 2 aproximadamente a la detecci n
137         # del filtro 3; 3 es proximo a 2
138         if beta23[i]<betaLim23 and beta34[i]<betaLim34:
139             print("    i+1 {0:d} case 1 beta23 {1:f} beta34 {2:f}".format(i+1,beta23[
120             i],beta34[i]))
140             flag[i]= 1
141             formula[i]=1
142             listaLABS.append(i+1)
143             listaLABScaso1.append(i+1)
144
145             # caso 3: detecci n del filtro 2 mayor que lo detectado
146             # por el filtro 3; 2 es mayor que 3
147             elif alpha23[i]>alphaLim23 and beta34[i]<betaLim34:
148                 print("    i+1 {0:d} case 3 beta23 {1:f} beta34 {2:f}".format(i+1,beta23[
121                 i],beta34[i]))
149                 flag[i]=2
150                 formula[i]=3
151                 listaLAES.append(i+1)
152                 listaLAEScaso3.append(i+1)
153             #-----
154             ## Fuente SI detectada en filtros 1, 2 y 3.
155             #-----
156             elif flujo[0]>0 and flujo[1]>0 and flujo[2]>0 and flujo[3]>0:
157                 # Si se detect en 1,2 y 3
158                 # caso 2:

```

```

159         if gamma21[i]<gammaLim21 and beta23[i]<betaLim23 and beta34[i]<betaLim34:
160             print("      i+1 {0:d} case 2 X21 {1:f} eX21 {2:f} beta23 {3:f} beta34 {4:f}
161             }".format(i+1,X21[i],eX21[i],beta23[i],beta34[i]))
162             # Si  $2 < F2/F1 < 8$ 
163             flag[i]=1
164             formula[i]=2
165             listaLABS.append(i+1)
166             listaLABScaso2.append(i+1)
167             # caso 4: cociente entre el flujo del filtro 3 entre el del flujo 1
168             # comprendido entre dos y ocho al tiempo que lo que detecta el filtro 2
169             # es mayor que lo detectado por el filtro 3
170             elif gamma31[i]<gammaLim31 and (alpha23[i]>alphaLim23) and beta34[i]<
betaLim34:
171                 print("      i+1 {0:d} case 4 X31 {1:f} eX31 {2:f} beta23 {3:f} beta34 {4:f}
172                 }".format(i+1,X31[i],eX31[i],beta23[i],beta34[i]))
173                 # Si  $2 < F3/F1 < 8$ 
174                 flag[i]=2 #se tiene una LAE
175                 formula[i]=4
176                 listaLAES.append(i+1)
177                 listaLAEScaso4.append(i+1)
178             #-----
179             ## Fuente no clasificable.
180             #-----
181             else:
182                 print("      i {0:d} case 0 ".format(i+1))
183                 flag[i]=0 #no se detecta ni LAE ni LAB
184                 formula[i]=0
185
186             #-----
187             ### Escribe tabla con los resultados de las operaciones realizadas
188             ### para las condiciones para todas las fuentes:
189             #-----
190
191             archOutResult = open("calculosCondiciones.txt","w")
192             # Pone la cabecera del archivo de salida
193             archOutResult.write("#Index"+"      "+"formula"+"      "+"alpha23"+"      "+"beta23"+"      "+"
beta34"+"      "+"gamma12"+"      "+"gamma13"+"\\n")
194             # Escribe en un solo archivo de salida.
195             for j in range(len(formula)):
196                 if formula[j]>0:
197                     archOutResult.write(str(int(index[j]))+"      "+str(formula[j])+"      "+str(round(
alpha23[j],3))+"      "+str(round(beta23[j],3))+"      "+str(round(beta34[j],3))+"      "+
str(round(gamma21[j],3))+"      "+str(round(gamma31[j],3))+"      "+"\\n")
198
199             archOutResult.close()
200
201
202
203
204
205             #####
206             # REPRESENTACIONES #
207             #####
208

```

```

209 ##### Genero las representaciones por cada fuente.
210 # (Hacer if True si se quiere que el programa lea el for;
211 # hacer if false si no se quiere leer el for)
212 if False:
213     print("\n\n Starting plotting")
214     for i in range(f):
215         print("      Plotting i+1 "+str(i+1)+" index "+str(index[i]))
216         # Selecciona los flujos.
217         flujo = datos[i,3:16:2]
218         #print(flujo.shape)
219         # Selecciona los errores de los flujos.
220         errorFlujo = datos[i,4:17:2]
221         # No abre las imagenes creadas
222         if (flujo[0]<0):
223             print("      ### F1<0 in i+1 "+str(i+1)+" index "+str(index[i]))
224
225         plt.ioff()
226         # Reserva una ventana para cada imagen
227         plt.figure(i)
228         # Selecciona los flujos.
229         flujo = datos[i,3:16:2]
230         # Selecciona los errores de los flujos.
231         errorFlujo = datos[i,4:17:2]
232         #-----
233         ## Grafica.
234         #-----
235         mpl.rc('xtick',top=True)
236         mpl.rc('ytick',right=True)
237         plt.errorbar(longOnd,flujo,yerr=errorFlujo,ecolor='black',fmt='.',marker='o',
markersize=4,mfc='red',mew=0)
238         plt.ylim(min(min(flujo-errorFlujo),0),max(flujo+errorFlujo))
239         plt.ylabel(r'$F (10^{-17}\mathrm{erg} \mathrm{cm}^{-2} \mathrm{s}^{-1} \mathrm{AA}^{-1})$')
240         plt.xlabel(r'$\lambda (\mathrm{nm})$')
241         #Dibuja una linea discontinua marcando el nivel del cero.
242         plt.axhline(y=0,xmin=0,xmax=744,linestyle='--')
243         #-----
244         ## Modifica el tama o de la imagen de salida.
245         #-----
246         fig = plt.gcf()
247         # Modifico las dimensiones con el factor de conversi n:
248         # 2 inches = 144 pix.
249         # Primer argumento es el ancho; segundo argumento es la altura.
250         fig.set_size_inches(700*2/200,425*3/300)
251         # Guarda las imagenes
252         plt.savefig('/home/maria/Escritorio/TFGcodigos/consola/
graficasCalibradasConEstrella2/'+str(i+1).zfill(4)+' .png')
253         plt.close('all')
254 #-----
255 ### Escribe un fichero para ser leído en Topcat, con la informacion
256 ### de cada fuente, la direccion de las finding charts y la de las
257 ### graficas de los flujos calibrados.
258 #-----
259
260 outFile = open(galaxiasTipo+'.topcat','w')
261 outFile.write("# index"+" "+"AsR"+" "+"Dec"+" "+"F1"+" "+"EF1"+" "+"F2"+" "+"
EF2"+" "+"F3"+" "+"EF3"+" "+"F4"+" "+"EF4"+" "+"F5"+" "+"EF5"+" "+"F6"+" "+"

```

```

    +"EF6"+" "+"F7"+" "+"EF7"+" "+"Flag"+" "+"Formula"+" "+"FigureGraficas"+" "+"
    FotosJuntoGraficas"+"\\n")
262
263
264 for i in range(f):
265     outFile.write(str(int(datos[i,0]))+" "+str(datos[i,1]))+" "+str(datos[i,2]))+" "
    +str(datos[i,3]))+" "+str(datos[i,4]))+" "+str(datos[i,5]))+" "+str(datos[i,6]))+"
    "+str(datos[i,7]))+" "+str(datos[i,8]))+" "+str(datos[i,9]))+" "+str(datos[i,10]))
    +" "+str(datos[i,11]))+" "+str(datos[i,12]))+" "+str(datos[i,13]))+" "+str(datos[
    i,14]))+" "+str(datos[i,15]))+" "+str(datos[i,16]))+" "+str(int(flag[i]))+" "+str
    (int(formula[i]))+" "+"/home/maria/Escritorio/TFGcodigos/consola/
    graficasCalibradasConEstrella2/"+str(int(datos[i,0])).zfill(4)+".png"+" "+"/home/
    maria/Escritorio/TFGcodigos/consola/findingChartsYgraficasCalibradas2/
    asociadaFindingFuente"+str(int(datos[i,0]))+".png"+"\\n")
266 #-----
267 ### Resultados mostrados por pantalla.
268 #-----
269 print(datos[0,0])
270 print(np.unique(flag))
271 print(np.unique(formula,return_counts=True))
272 if np.any(flag==2):
273     print('N mero de LAEs que hay en total: ',np.unique(flag,return_counts=True)
    [1][2])
274 else:
275     print('No se selecciona ninguna galaxia como LAE')
276
277 if np.any(flag==1):
278     print('N mero de LABs que hay en total: ',np.unique(flag,return_counts=True)
    [1][1])
279 else:
280     print('No se selecciona ninguna galaxia como LAB.')
281
282 print('N mero de LABs caso1 que hay en total: ',np.unique(formula,return_counts=True)
    [1][1])
283 print('N mero de LABs caso2 que hay en total: ',np.unique(formula,return_counts=True)
    [1][2])
284
285 if np.any(formula==3) and np.any(formula==4):
286     print('N mero de LAEs caso3 que hay en total: ',np.unique(formula,return_counts=
    True)[1][3])
287     print('N mero de LAEs caso4 que hay en total: ',np.unique(formula,return_counts=
    True)[1][4])
288 else:
289     print('No se selecciona ninguna galaxia como LAE')
290
291 if np.any(formula==4) and (not np.any(formula==3)):
292     print('N mero de LAEs caso4 que hay en total: ',np.unique(formula,return_counts=
    True)[1][3])
293 else:
294     print('No se selecciona ninguna galaxia como LAE del caso 4')
295
296 if np.any(formula==3) and (not np.any(formula==4)):
297     print('N mero de LAEs caso3 que hay en total: ',np.unique(formula,return_counts=
    True)[1][3])
298 else:
299     print('No se selecciona ninguna galaxia como LAE del caso 3')

```

```
300
301 print('N mero de fuentes que no han sido clasificados como LAEs o LABs que hay en
      total: ',np.unique(flag,return_counts=True)[1][0])
302 print('Fuentes de LAEs detectadas (las cifras se corresponden con el index del
      archivo de flujoscalibrados): ',listaLAES)
303 print('Fuentes de LABs detectadas (las cifras se corresponden con el index del
      archivo de flujoscalibrados): ',listaLABS)
304
305 print('Fuentes de LABs caso 1 detectadas (las cifras se corresponden con el index del
      archivo de flujoscalibrados): ',listaLABScaso1)
306 print('Fuentes de LABs caso 2 detectadas (las cifras se corresponden con el index del
      archivo de flujoscalibrados): ',listaLABScaso2)
307 print('Fuentes de LABs caso 3 detectadas (las cifras se corresponden con el index del
      archivo de flujoscalibrados): ',listaLAEScaso3)
308 print('Fuentes de LABs caso 4 detectadas (las cifras se corresponden con el index del
      archivo de flujoscalibrados): ',listaLAEScaso4)
309
310 outFile.close()
311
312 ## Fin metodoSeleccion.py -----
```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on May 2019
5 @author: Maria Ruiz Perez
6 """
7
8 #####
9 ##          PR10          Nombre:asociaFindingChartsFuentes          ##
10 ##-----##
11 ## Programa que asociada cada findding chart con la grafica de la fuente          ##
12 ## correspondiente.          ##
13 ## Requiere:          ##
14 ##          -Finding charts creadas.          ##
15 ##          -Graficas de fuentes calibradas creadas.          ##
16 #####
17
18 import numpy as np
19 import cv2 # para importar el paquete opencv para python
20
21 # (!)Instalar open cv antes de ejecutar
22 # pip install opencv-python
23
24 numFotos = 986 # Numero de parejas entre graficas y fotos que va a juntar
25 for i in range(numFotos):
26     print(i)
27     # Nombre imagen 1
28     imageFile1 = '/home/maria/Escritorio/TFGcodigos/consola/findingChartsDS9/foto'+
29     str(i+1).zfill(4)+'.jpg'
30     # Nombre imagen 2 graficas
31     imageFile2 = '/home/maria/Escritorio/TFGcodigos/consola/
32     graficasCalibradasConEstrella2/'+str(i+1).zfill(4)+'.png'
33     # Lee imagen 1
34     img1 = cv2.imread(imageFile1)
35     # Lee imagen 2
36     img2 = cv2.imread(imageFile2)
37     # Dimensiones imagen 1
38     h1, w1 = img1.shape[:2]
39     print(img1.shape)
40     # Dimensiones imagen 2
41     h2, w2 = img2.shape[:2]
42     print(img2.shape)
43     # Genera algo similar a una matriz grande.
44     # Crea una matriz vacia; altura maxima entre h1 o h2; el 3 viene a que realmente
45     # es una matriz tridimensional;
46     # uint es el formato con el que se guarda la informacion de la imagen
47     vis = np.zeros((max(h1, h2), w1+w2,3), np.uint8)
48     # Combina dos imagenes
49     vis[:h1,:w1,:3] = img1
50     # Ahora la segunda imagen.
51     vis[:h2,w1:w1+w2,:3] = img2
52     # Genera la imagen.
53     cv2.imwrite('/home/maria/Escritorio/TFGcodigos/consola/
54     findingChartsYgraficasCalibradas2/'+str(i+1)+'.png', vis)
55     print(' ')
56

```

```

53 ## Fin asociaFindingChartFuente.py -----
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  """
5  Created on 9 June 2019
6  author: Maria Ruiz Perez
7
8  """
9  #####
10 ##          PR 11          Nombre: diagramasCriteriosFisicos.py          ##
11 ##-----##
12 ## Programa en el que se crean diagramas regidos por los criterios fisicos ##
13 ## con los que se crea en otro programa el metodo de seleccion.          ##
14 ## Los diagramas son una explicacion ilustrativa de los criterios fisicos. ##
15 #####
16
17 import matplotlib.pyplot as plt
18 import numpy as np
19
20 rutaSalida = "/home/maria/Escritorio/TFGcodigos/consola/diagramasLBGyLAE/"
21 #-----
22 ### Caso 1.
23 #-----
24 # Funcion escalon:
25 ejeXCaso1Escalon = np.array([735, 738, 738, 741])
26 ejeYCaso1Escalon = np.array([0.0, 0.0, 0.1, 0.1])
27 # Puntos:
28 ejeXCaso1 = np.array([736,738, 739, 740])
29 ejeYCaso1 = np.array([0.0, 0.1, 0.09, 0.11])
30
31 plt.figure(1)
32 # Pone ticks a todos los ejes.
33 plt.rc('xtick',top=True)
34 plt.rc('ytick',right=True)
35 # Grafica.
36 plt.plot(ejeXCaso1Escalon,ejeYCaso1Escalon,'b')
37 plt.errorbar(ejeXCaso1,ejeYCaso1,yerr=0.01,c='r',mfc='r',ms=10.0,marker='.',linestyle
38             ='None', capsize=10)
39 plt.xlabel(r'$\lambda$ (nm)$',fontsize='large')
40 plt.ylabel(r'$F_{\lambda}(10^{-17}\mathrm{erg}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{AA}^{-1})$',fontsize='
41             large')
42 plt.text(735.8,0.004,'1',fontsize='medium')
43 plt.text(737.8,0.103,'2',fontsize='medium')
44 plt.text(738.7,0.09,'3',fontsize='medium')
45 plt.text(739.7,0.114,'4',fontsize='medium')
46
47 # Limites ejes.
48 plt.ylim(-0.0125,0.14)
49 plt.xlim(735,741)
50 # Guarda la figura.
51 plt.savefig(rutaSalida+"caso1LBG"+" .jpg", bbox_inches='tight')
52 plt.show()
53
54 #-----
55 ### Caso 2.

```

```

54 #-----
55
56 # Funcion escalon.
57 ejeXCaso1Escalon = np.array([735, 738, 738, 741])
58 ejeYCaso1Escalon = np.array([0.03, 0.03, 0.1, 0.1])
59 # Puntos.
60 ejeXCaso1 = np.array([736,738, 739, 740])
61 ejeYCaso1 = np.array([0.03, 0.1, 0.09, 0.11])
62
63 plt.figure(2)
64 # Pone ticks a todos los ejes.
65 plt.rc('xtick',top=True)
66 plt.rc('ytick',right=True)
67 # Grafica.
68 plt.plot(ejeXCaso1Escalon,ejeYCaso1Escalon,'b')
69 plt.errorbar(ejeXCaso1,ejeYCaso1,yerr=0.01,c='r',mfc='r',ms=10.0,marker='.',linestyle
    ='None', capsize=10)
70 plt.xlabel(r'$\lambda$ (nm)$',fontsize='large')
71 plt.ylabel(r'$F_{\lambda}(10^{-17}\mathrm{erg}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{AA}^{-1})$',fontsize='
    large')
72 plt.text(735.6,0.065,'Escal n',fontsize='medium')
73 plt.text(735.8,0.035,'1',fontsize='medium')
74 plt.text(737.8,0.103,'2',fontsize='medium')
75 plt.text(738.75,0.09,'3',fontsize='medium')
76 plt.text(739.7,0.11,'4',fontsize='medium')
77 # Flecha escalon.
78 plt.annotate(s='', xy=(1,1), xytext=(0,0), arrowprops=dict(arrowstyle='<->'))
79 plt.arrow(735.5,0.03,0,0.07,head_width=0.05, head_length=0.003, linewidth=1,
    length_includes_head=True)
80 plt.arrow(735.5,0.1,0.0,-0.07,head_width=0.05, head_length=0.003, linewidth=1,
    length_includes_head=True)
81 # Linea discontinua.
82 linDiscX = np.array([735,738])
83 linDiscY = np.array([0.1, 0.1])
84 plt.plot(linDiscX,linDiscY,linestyle='--')
85 # Limites ejes.
86 plt.ylim(0,0.14)
87 plt.xlim(735,741)
88 # Guarda.
89 plt.savefig(rutaSalida+"caso2LBG"+" .jpg", bbox_inches='tight')
90 plt.show()
91
92
93 #-----
94 ### Caso 3.
95 #-----
96
97 # Funcion escalon:
98 ejeXCaso1Escalon = np.array([735,737.7,738, 738.2, 741])
99 ejeYCaso1Escalon = np.array([0.0,0.0 ,0.145, 0.1, 0.1])
100 # Puntos:
101 ejeXCaso1 = np.array([736, 738, 739, 740])
102 ejeYCaso1 = np.array([0.0, 0.145, 0.1, 0.12])
103
104 plt.figure(3)
105 # Pone ticks a todos los ejes.

```

```

106 plt.rc('xtick',top=True)
107 plt.rc('ytick',right=True)
108 # Grafica.
109 plt.plot(ejeXCaso1Escalon,ejeYCaso1Escalon,c='b')
110 plt.errorbar(ejeXCaso1,ejeYCaso1,yerr=0.01,c='r',mfc='r',ms=10.0,marker='.',linestyle
    ='None', capsize=10)
111 plt.xlabel(r'$\lambda$ (nm)$',fontsize='large')
112 plt.ylabel(r'$F_{\lambda}(10^{-17}\mathrm{erg}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{\AA}^{-1})$',fontsize='
    large')
113 # Texto
114 plt.text(735.7,0.005,'1',fontsize='medium')
115 plt.text(737.8,0.145,'2',fontsize='medium')
116 plt.text(738.7,0.104,'3',fontsize='medium')
117 plt.text(739.8,0.12,'4',fontsize='medium')
118 # Flecha escalon
119 plt.text(738.2,0.125,'Pico emisi n',fontsize='medium')
120 plt.arrow(739.5,0.1,0,0.045,head_width=0.05, head_length=0.003, linewidth=1,
    length_includes_head=True)
121 plt.arrow(739.5,0.145,0.0,-0.045,head_width=0.05, head_length=0.003, linewidth=1,
    length_includes_head=True)
122 # Linea discontinua salto.
123 linDiscX = np.array([738,739.5])
124 linDiscY = np.array([0.145, 0.145])
125 plt.plot(linDiscX,linDiscY,linestyle='--',c='orange')
126 # Fija limite ejeX.
127 plt.xlim(735,741)
128
129 # Guarda la figura.
130 plt.savefig(rutaSalida+"caso3LBE"+" .jpg", bbox_inches='tight')
131 plt.show()
132
133
134 #-----
135 ### Caso 4.
136 #-----
137
138 # Funcion escalon.
139 ejeXCaso1Escalon = np.array([735,737.7,738, 738.2, 741])
140 ejeYCaso1Escalon = np.array([0.03,0.03 ,0.145, 0.1, 0.1])
141 # Puntos.
142 ejeXCaso1 = np.array([736, 738, 739, 740])
143 ejeYCaso1 = np.array([0.03, 0.145, 0.1, 0.12])
144
145 plt.figure(4)
146 # Pone ticks a todos los ejes.
147 plt.rc('xtick',top=True)
148 plt.rc('ytick',right=True)
149 # Grafica.
150 plt.plot(ejeXCaso1Escalon,ejeYCaso1Escalon,'b')
151 plt.errorbar(ejeXCaso1,ejeYCaso1,yerr=0.01,c='r',mfc='r',ms=10.0,marker='.',linestyle
    ='None', capsize=10)
152 plt.xlabel(r'$\lambda$ (nm)$',fontsize='large')
153 plt.ylabel(r'$F_{\lambda}(10^{-17}\mathrm{erg}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{\AA}^{-1})$',fontsize='
    large')
154 plt.text(735.6,0.06,'Escal n',fontsize='medium')
155 plt.text(735.7,0.035,'1',fontsize='medium')

```

```
156 plt.text(737.8,0.143,'2',fontsize='medium')
157 plt.text(738.8,0.105,'3',fontsize='medium')
158 plt.text(739.7,0.12,'4',fontsize='medium')
159 # Flecha break.
160 plt.annotate(s='', xy=(1,1), xytext=(0,0), arrowprops=dict(arrowstyle='<->'))
161 plt.arrow(735.5,0.03,0,0.07,head_width=0.05, head_length=0.003, linewidth=1,
           length_includes_head=True)
162 plt.arrow(735.5,0.1,0.0,-0.07,head_width=0.05, head_length=0.003, linewidth=1,
           length_includes_head=True)
163 # Flecha salto pico de emisi n.
164 plt.text(738.2,0.125,'Pico emisi n',fontsize='medium')
165 plt.arrow(739.5,0.1,0,0.045,head_width=0.05, head_length=0.003, linewidth=1,
           length_includes_head=True)
166 plt.arrow(739.5,0.145,0.0,-0.045,head_width=0.05, head_length=0.003, linewidth=1,
           length_includes_head=True)
167 # Linea discontinua break
168 linDiscX = np.array([735,738])
169 linDiscY = np.array([0.1, 0.1])
170 plt.plot(linDiscX,linDiscY,linestyle='--')
171 plt.xlim(735,741)
172 # Linea discontinua salto.
173 linDiscX = np.array([738,739.5])
174 linDiscY = np.array([0.145, 0.145])
175 plt.plot(linDiscX,linDiscY,linestyle='--')
176 # Guarda la figura.
177 plt.savefig(rutaSalida+"caso4LBE"+" .jpg", bbox_inches='tight')
178 plt.show()
179
180
181 ## Fin diagramasCriteriosFisicos.py -----
```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Representa histogramas
5 @Author: Maria Ruiz Perez
6 @Fecha: Junio 2019
7 """
8
9 #####
10 ##          PR 12          Nombre: histogramas.py          ##
11 ##-----##
12 ## Programa que crea histogramas para ilustrar los calculos realizados con ##
13 ## el metodo de seleccion. ##
14 ## Requiere: ##
15 ##     Archivo de salida del metodoSeleccion.py ##
16 ## Retorna: ##
17 ##     Cinco histogramas. Cada histograma se corresponde con una columna de ##
18 ##     resultados del archivo de entrada. ##
19 #####
20
21 import glob
22 import numpy as np
23 import matplotlib.pyplot as plt
24 from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, AutoMinorLocator)
25
26 # Nombres de los ficheros de entrada
27 name = glob.glob('calculosCondiciones.txt') # glob es una funcion del paquete glob.
28     # permite guardar todos los nombres de un conjunto de ficheros
29     # por ejemplo 1000 ficheros .txt con nombres
30     # diferentes. Los guarda de golpe sin necesidad de guardar uno por
31     # uno. En este caso innecesario su uso porque solo
32     # tengo un fichero. Se ha usado por aprender una funcion mas.
33
34 # Lee las columnas que interesan del fichero
35 data = np.genfromtxt(name[0], skip_header=1, usecols=(2,3,4,5,6))
36 # Crea listas vacias
37 alpha23= []
38 beta23 = []
39 beta34 = []
40 gamma21 = []
41 gamma13 = []
42
43 # Rellena las listas con la informacion de data
44 for i in range(len(data)):
45     #solo permite guardar numeros en las listas vacias
46     #condicion: si el elemento de la casilla iesima no es un numero
47     #! diferente.
48     # Solo se aplica el if a las columnas de gamma por ser las unicas que
49     # que presentan casillas con nan.
50     alpha23.append(data[i,0])
51     beta23.append(data[i,1])
52     beta34.append(data[i,2])
53     if str(data[i,3]) != 'nan':
54         gamma21.append(data[i,3])
55     else:
56         gamma21.append(-1)

```

```

54
55     if str(data[i,4])!='nan':
56         gamma13.append(data[i,4])
57     else:
58         gamma13.append(-1)
59
60 # Ruta de salida
61 rutaSalida = "/home/maria/Escritorio/TFGcodigos/consola/histogramas/"
62
63 #####
64 ##      REPRESENTACIONES DE HISTOGRAMAS      ##
65 #####
66 #-----
67 ### Representa histograma de alpha_2,3
68 #-----
69 plt.figure(1)
70 # Pone ticks a todos los ejes.
71 plt.rc('xtick',top=True)
72 plt.rc('ytick',right=True)
73 # Ejes.
74 axes=plt.axes()
75 # Pone ticks menores a los ejes.
76 axes.xaxis.set_minor_locator(MultipleLocator(0.1))
77 axes.yaxis.set_minor_locator(MultipleLocator(0.5))
78 # Grafica.
79 n,bins,p = plt.hist(alpha23,bins=15,facecolor='red',alpha=0.75,hatch="//",edgecolor='
    red')
80 # Numero de datos.
81 label = [r'$N=$'+str(int(sum(n)))]
82 # Leyenda.
83 plt.legend(label,fontsize='medium')
84 # Eje X
85 plt.xlabel(r'$\alpha_{2,3}$',fontsize='large')
86 # Eje Y
87 plt.ylabel('Frecuencia',fontsize='large')
88 # Fija los limites en el eje Y que se quieren (el ultimo valor no lo escribe).
89 y = range(0,16,2)
90 plt.yticks(y)
91 # Guarda.
92 plt.savefig(rutaSalida+"alpha23"+" .jpg", bbox_inches='tight')
93 #-----
94 ### Representa histograma de beta_2,3
95 #-----
96 plt.figure(2)
97 #Pone ticks a todos los ejes
98 plt.rc('xtick',top=True)
99 plt.rc('ytick',right=True)
100 # Ejes.
101 axes=plt.axes()
102 # Pone ticks menores a los ejes.
103 axes.xaxis.set_minor_locator(MultipleLocator(0.1))
104 axes.yaxis.set_minor_locator(MultipleLocator(0.5))
105 # Grafica.
106 n,bins,p = plt.hist(beta23,bins=15,facecolor='blue',alpha=0.75,hatch="//",edgecolor='
    blue')
107 # Numero de datos.

```

```

108 label = [r'$N=$'+str(int(sum(n)))]
109 # Leyenda.
110 plt.legend(label,fontsize='medium')
111 # Eje X
112 plt.xlabel(r'$\beta_{2,3}$',fontsize='large')
113 # Eje Y
114 plt.ylabel('Frecuencia',fontsize='large')
115 # Fija los limites en el eje Y que se quieren (el ultimo valor no lo escribe).
116 y = range(0,20,2)
117 plt.yticks(y)
118 # Guarda.
119 plt.savefig(rutaSalida+"beta23"+" .jpg", bbox_inches='tight')
120
121
122 #-----
123 ### Representa histograma beta_3,4
124 #-----
125 plt.figure(3)
126 # Pone ticks a todos los ejes.
127 plt.rc('xtick',top=True)
128 plt.rc('ytick',right=True)
129 # Ejes.
130 axes=plt.axes()
131 # Pone ticks menores a los ejes.
132 axes.xaxis.set_minor_locator(MultipleLocator(0.05))
133 axes.yaxis.set_minor_locator(MultipleLocator(0.5))
134 # Grafica.
135 n,bins,p = plt.hist(beta34,bins=10,facecolor='orange',alpha=0.75,hatch="//",edgecolor
    ='black')
136 # Numero de datos.
137 label = [r'$N=$'+str(int(sum(n)))]
138 # leyenda.
139 plt.legend(label,fontsize='medium')
140 # Eje X
141 plt.xlabel(r'$\beta_{3,4}$',fontsize='large')
142 # Eje Y
143 plt.ylabel('Frecuencia',fontsize='large')
144 # Fija los limites en el eje Y que se quieren (el ultimo valor no lo escribe).
145 y = range(0,14,2)
146 plt.yticks(y)
147 # Guarda.
148 plt.savefig(rutaSalida+"beta34"+" .jpg", bbox_inches='tight')
149
150
151 #-----
152 ### Representa histograma gamma_1,2
153 #-----
154 ## Para tener solo en cuenta los valores mayores o iguales a cero: [gamma21>=0]
155 ## Este comando solo es valido para arrays; convierte la lista gamma21 en array
156 gamma21array = np.asarray(gamma21)
157 plt.figure(4)
158 # Pone ticks a todos los ejes.
159 plt.rc('xtick',top=True)
160 plt.rc('ytick',right=True)
161 # Ejes.
162 axes=plt.axes()

```

```

163 # Pone ticks menores a los ejes.
164 axes.xaxis.set_minor_locator(MultipleLocator(0.05))
165 axes.yaxis.set_minor_locator(MultipleLocator(0.5))
166 # Grafica.
167 n,bins,p = plt.hist(gamma21array[gamma21array>=0],bins=10,facecolor='black',alpha
    =0.75,hatch="//",edgecolor='black')
168 # Numero de datos
169 x=gamma21array[gamma21array>=0]
170 label = [r'$N=$'+str(len(x))]
171 # leyenda
172 plt.legend(label,fontsize='medium')
173 # Eje X
174 plt.xlabel(r'$\gamma_{1,2}$',fontsize='large')
175 # Eje Y
176 plt.ylabel('Frecuencia',fontsize='large')
177 # Fija los limites en el eje Y que se quieren (el ultimo valor no lo escribe).
178 y = range(0,20,2)
179 plt.yticks(y)
180 # Guarda.
181 plt.savefig(rutaSalida+"gamma12"+" .jpg", bbox_inches='tight')
182
183 #-----
184 ### Representa histograma gamma_1,3
185 #-----
186 gamma13array = np.asarray(gamma13)
187 plt.figure(5)
188 # Pone ticks a todos los ejes.
189 plt.rc('xtick',top=True)
190 plt.rc('ytick',right=True)
191 # Ejes.
192 axes=plt.axes()
193 # Pone ticks menores a los ejes.
194 axes.xaxis.set_minor_locator(MultipleLocator(0.1))
195 axes.yaxis.set_minor_locator(MultipleLocator(0.5))
196 n,bins,p = plt.hist(gamma13array[gamma13array>=0],bins=10,facecolor='green',alpha
    =0.75,hatch="//",edgecolor='black')
197 #plt.ylim(0,18)
198 # Numero de datos.
199 q=gamma13array[gamma13array>=0]
200 label = [r'$N=$'+str(len(q))]
201 # Legenda.
202 plt.legend(label,fontsize='medium')
203 # Eje X
204 plt.xlabel(r'$\gamma_{1,3}$',fontsize='large')
205 # Eje Y
206 plt.ylabel('Frecuencia',fontsize='large')
207 # Fija los limites en el eje Y que se quieren.(el ultimo valor no lo escribe).
208 y = range(0,22,2)
209 plt.yticks(y)
210 # Guarda.
211 plt.savefig(rutaSalida+"gamma13"+" .jpg", bbox_inches='tight')
212
213 ## Fin histogramas.py -----

```

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on 24 May 2019.
5 @author: Maria Ruiz Perez
6 Unidades en el sistema cgs
7 """
8 #####
9 ##          PR13          Nombre: superponeEstudios.sh          ##
10 ##-----##
11 ## Abre el ds9 con las .reg del estudio de Pridley y la de los          ##
12 ## casos 1 2 y 4 del TFG.          ##
13 #####
14 ds9 m0s_8.fits -geometry 1000x1000 -frame 1 -wcs align yes -match frames wcs -frame 1
    -invert -region load regPridley.reg -region load listaLAB_y_LAE.reg -scale linear
    -scale mode zscale -scale scope local -zoom to 2 -view colorbar no -saveimage jpg
    fotoNew0920.jpg #-quit

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on May 2019
5 @author: Maria Ruiz Perez.
6 """
7 #####
8 ##          PR14          Nombre: fuentesASuperponer.py          ##
9 ##-----##
10 ## Programa que genera regiones sobre las fuentes de tipo LAB y LAE          ##
11 ## detectadas en el fichero galaxiasTipo.topcat          ##
12 ## Requiere:          ##
13 ##          -Lista de regiones de todas las fuentes (archivo retornado          ##
14 ##          en metodoSeleccion.py)          ##
15 ## Retorna:          ##
16 ##          -Archivo .reg con regiones de fuentes seleccionadas          ##
17 ##          marcadas a color.          ##
18 #####
19
20 # Importa paquetes matematicos
21 import numpy as np
22
23 # Fichero de datos.
24 nameFile = 'galaxiasTipo.topcat'
25 data = np.loadtxt(nameFile, usecols = (0,1,2,17))
26 #-----
27 ### Extrae informacion.
28 #-----
29 # Indice
30 index = data[:,0]
31 # Ascension Recta
32 ascRec = data[:,1]
33 # Declinacion
34 declin = data[:,2]
35 # Flag
36 flag = data[:,3]
37
38
39 ## Genera un fichero con regiones LAB y LAE

```

```
40
41 outFile = open('listaLAB_y_LAE.reg','w')
42
43 for i in range(len(flag)):
44     if flag[i]==1:
45         outFile.write('fk5;circle('+str(ascRec[i])+','+str(declin[i])+',2")'
46                        +'# color=green text={' +str(index[i])+'};'+'\n')
47     elif flag[i]==2:
48         outFile.write('fk5;circle('+str(ascRec[i])+','+str(declin[i])+',2")'
49                        +'# color=red text={' +str(index[i])+'};'+'\n')
50     print(i)
51 outFile.close()
52
53 ## Fin fuentesAsuperponer.py-----
```